



★ 本期焦点

下一代安全研究模型及发展趋势

手机银行的几个典型风险

几个常见的
DDoS botnet及其特点

《个人网上银行登录安全研究报告》提要

本期看点 HEADLINES

3 下一代安全研究模型及发展趋势

35 手机银行的几个典型风险

47 几个常见的DDoS botnet及其特点

63 《个人网上银行登录安全研究报告》摘要



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 8880-8667
传真：(010)6872 8708
网址：www.nsfocus.com

2013/07 总第 021

Nsmagazine@nsfocus.com

安全+ SECURITY+

© 2013 绿盟科技

本刊图片与文字未经相关版权所有人书面批准，一概不得以任何形式、方法转载或使用。本刊保留所有版权。

SECURITY+ 是绿盟科技的注册商标。

需要获取更多信息，请访问WWW.NSFOCUS.COM

卷首语	赵粮	2
专家视角		3-22
下一代安全研究模型及发展趋势	李鸿培	3
从“软件定义网络”到“软件定义安全”	王卫东	9
数据中心 Web 系统协同防护思路探讨	田民	16
行业热点		23-46
网络结构安全在电子政务中的应用 ——两种典型网络结构的安全对比分析	张智南	23
水煮互联网——漫谈用户认证	罗爱国	27
手机银行的几个典型风险	蔡昆	35
银行信息安全管理探讨(二)	徐一丁	43
前沿技术		47-62
几个常见的 DDoS botnet 及其特点	刘亚 周大	47
Windows 驱动常见漏洞分析	董阳	53
Z4root 剖析	刘永军	58
《个人网上银行登录安全研究报告》提要	赵波 李哲祎 刘凯	63
综合信息		70-71
安全公告		72-80
NSFOCUS 2013 年 2-4 月之十大安全漏洞		72

新的威胁和机遇在哪里？

教皇在使用 iPad 上网，叙利亚反对派武装使用游戏机和电视等自制控制器驾驶土制装甲车，

他们还使用智能手机远程控制火箭炮，以 Vupen 为代表的“黑客”在叫卖网际军火……

这些都是在刚刚过去的 RSA 热点研讨会上绿盟科技研究院高级研究员于畅给大家展现的场景。

“黑色地下经济的存在，使得追逐经济利益已成为黑客及黑客组织发起有目标攻击的动力之一。

而政治、信仰、军事及商战的目的则造就了组织更为严密的攻击团体（网上恐怖主义、网络战），攻击国家关键信息基础设施（CII）、重要的工业控制系统以及窃取机密信息已成为组织严密的攻击团体的有目的的、持久攻击的对象”，这段话摘自绿盟科技研究院关于下一代安全的一份研究报告。

威胁在不断演化，“天下熙熙，皆为利来，天下攘攘，皆为利往”，金钱所在，目标所指。

网上银行和手机银行，伴随着网购的爆发性增长，成为非常普遍的个人金融工具，同时也引来了逐利攻击者的觊觎。

群狼环伺，何以对抗和自保？如何做到知己知彼，制定适当的防护战略？

网络技术也在不断演化。

“软件定义”的概念可能是起始于软件定义无线电 SDR，时过多年，软件定义并没有给产业界留下什么大的印象。

软件定义网络（SDN）本来也是一个由学术圈提出的实验室概念，可是却在短短的时间里风生水起，

不但得到了学术界和工业界的一起鼓掌，还衍生出了软件定义数据中心（SDDC）、

软件定义安全（SDSec）、软件定义 Internet 交换（SDX）、软件定义 Anything（SDx）等。

这里的软件定义并不是说原来的网络配置开通等都是硬件写死的，

而是说配置开通等控制平面的活动与交换转发等数据平面的活动分离开来，

可以更加方便地通过编程接口来实现，从而实现更大的开放、互操作、自动化。

SDN 带来的不仅仅是新技术所伴生的威胁，还给网络安全本身带来新的机遇，

Gartner 在其近期发布的一篇研究报告中就充满激情地拥抱了软件定义。新的威胁和机遇在哪里？

希望本期的十二篇文章能够给您带来共鸣和启发。

下一代安全研究模型及发展趋势

战略研究部 李鸿培

关键词：下一代安全 研究模型 发展趋势

摘要：本文在讨论网络安全发展趋势时，提出了一种基于关联角色能力变化的“下一代安全”分析研究模型。该模型能够通过分析当前的新安全问题和改进需求，对安全环境的变化所引起的针对安全防护能力的挑战及应对措施进行分析，进而推导总结出下一代安全的重点发展方向。

1. 引言

随着信息网络技术应用以及现实生活的日益融合，信息系统在涉及国计民生的各行各业日益发挥巨大的作用。因其重要性，这些信息系统面临来自互联网上潜在的黑客、竞争对手甚至是敌对势力的安全威胁也日益严重。系统攻击者的攻击动机也不仅是为了技术突破，而是更多地受到政治、经济、信仰等多方面的影响，攻击者基于共同的目标形成自发组织的攻击群体，甚至是形成组织严密的攻击团体。尤其是组织严密的攻击团体，能够拥有足够的经济、技术实力来开发新型的攻击手段，进而发起有组织、有目标的攻击。近年来，伊朗核电站被攻击、RSA 密钥泄密等安全事件表明：有组织、

有目的的危害极大的网络攻击事件正在迅速增多，而且攻击者多采用诸如 APT 的新型攻击手段、组合利用多个最新的 0-day 漏洞，使得传统的安全手段难以检测、防护。同时，云计算及虚拟化、大数据、移动互联网等新 IT 应用技术的快速发展，为用户提供了更为灵活、实用的 IT 应用及服务模式，但也为当前的信息安全产品及相应解决方案提出了新的挑战。

为了应对新的安全威胁及 IT 技术应用的挑战，业内提出了“下一代安全”的概念。但什么是“下一代安全”？“下一代安全产品”应该具有什么样的功能与特点？却一直没有一个明确的定义和论证。

这是因为“下一代”是一个很模糊的词，而且是一个有时效性

的动态概念。既然是“下一代”，那么必然是相对于“当代”而言。在目前缺乏对“当代安全”明确定义和统一认识的前提下，确定“下一代安全”与“当代安全”的“代差功能特征”无疑是一个非常困难的问题。而且关于“下一代安全”的说法自然也是众说纷纭，各个厂家基本上都是基于自己的技术积累、市场优势以及产品发展计划而规划、宣讲自己对下一代安全的理解。

在此背景下，我们对“下一代安全”的相关问题、需求进行了较为深入的研究，提出了关于“下一代安全”的研究模型和分析方法，并通过对比当前的安全问题和改进需求，分析推导出下一代安全的发展趋势及技术特征 [1]。

2. 下一代安全的研究模型

因为关于“下一代安全”发展趋势的预测，会对公司的技术与产品规划产生较大的影响，这就要求：在分析讨论“下一代安全”发展趋势的时候，必须强调充足的论据支持和严密的逻辑推导，同时要尽可能全面地考虑到各种可能的影响因素，尽可能保证分析预测结果的合理性、客观性与可信性。

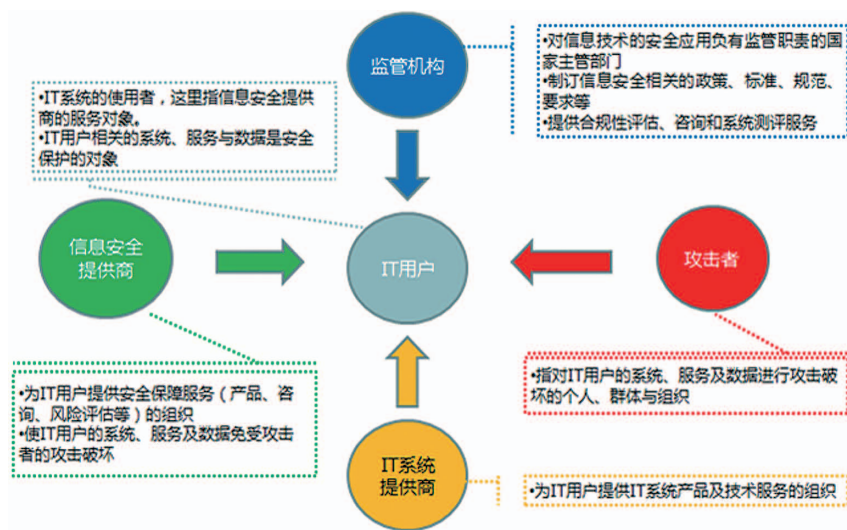


图 1 下一代安全研究的关联角色模型

2.1 下一代安全研究的关联角色模型

为保证在研究的过程中尽可能全面地考虑影响“下一代安全”的各种变化因素，我们提出了“下一代安全研究的关联角色模型”（如图 1 所示）。图 1 中定义了五种相关角色，即 IT 用户、IT 系统提供商、监管机构、攻击者与信息安全厂商。其中：

IT 用户：指 IT 系统的拥有者和使用者。IT 用户相关的 IT 系统、业务服务与数据是安全攻击的对象，同时也是信息安全服务的防护对象。

IT 系统提供商：指为 IT 用户提供 IT 系统（软件、硬件、网络及业务应用产品）及技术服务的组织。

监管机构：对信息技术的安全应用负有监管职责的国家 / 行业主管部门。制订信息安全

相关的政策、标准、规范、要求等，并提供合规性评估、系统评测及咨询服务等。

攻击者：指对 IT 用户的系统、服务及数据进行攻击破坏的个人、群体与团体组织。

信息安全厂商：为 IT 用户提供安全保障服务（安全产品、安全服务、咨询等）的组织，帮助 IT 用户抵御攻击者的攻击。

这五种角色是以攻防研究的对象（IT 用户及其 IT 系统）为中心来考虑的。进行“下一代安全”研究的目的是为了提高我们作为安全厂商的技术水平与安全服务能力。了解用户的安全需求、了解攻击者的能力及威胁、了解 IT 技术的最新发展以及监管部门的安全要求，都将有助于我们了解当前安全产品与技术服务能力的不足，有助于明确我们下一步进行安全服务能力提升的努力方向。显然，图 1 中各角色的技术与服务能力变化都可能改变 IT 用户系统的安全攻防态势，甚至对被保护的 IT 系统带来新的安全威胁，并对信息安全提供商的技术与服务能力提出新的挑战。

2.2 下一代安全研究的分析模型

基于上面的“下一代安全研究的关联角色模型”，我们进一步讨论了下一代安全研究的分析模型与相关术语概念，明确了“下一代安全研究”的逻辑分析流程（如图 2 所示）。

通过分析图 1 中各关联角色的技术能力和服务模式的变化，结合 IT 技术变革、市场需求及商业模式的演变、攻防技术的发展以及监管部门的要求等各种可能影响安全防护能力的因素，全面分析信息安全提供商所面临的挑战，并提出相应的应对策略。进而对各应

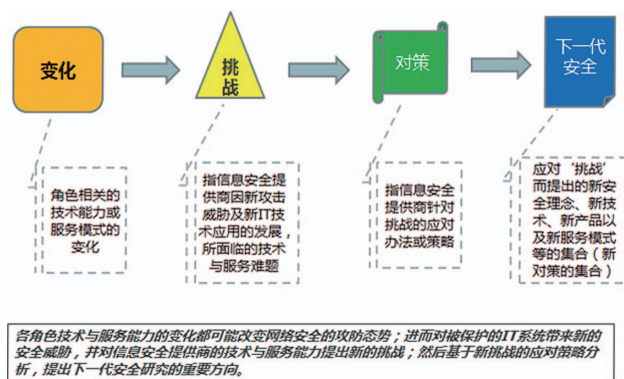


图 2 下一代安全研究的分析模型

对策略进行分类、统计，进而归纳总结出“下一代安全”的主要特征及发展趋势。

2.3 下一代安全的概念定义

基于上面的研究分析模型，针对“下一代安全”概念，我们给出了一个相对明确的理论定义。

定义 1. 下一代安全

下一代安全是指为应对因新的安全威胁与 IT 技术的发展而造成安全技术水平及安全服务能力严重不足的问题（挑战），所提出的新安全理念、技术、产品以及服务模式等对策的集合 [1]。

由于“下一代安全研究的分析模型”强调的是对攻防环境内各种变化因素的综合分析和逻辑推导，显然分析得到的关于未来一段时间内所需要提升的安全能力和技术发展趋势将更符合实际需求，以此为基础所确定的下一代安全的“代差特征”也将更容易得到业

内的共识。

3. 下一代安全的发展趋势

基于下一代安全的研究分析模型，我们从攻击者、IT 系统提供商、IT 用户、监管机构以及信息安全提供商等相关角色的不同视角，对各角色所面对的技术能力与服务模式的发展变化以及这些变化对未来信息安全防护所带来的挑战及应对策略进行了详细的分析讨论，并对得到的各种应对策略进行统计与归类分析（本文因篇幅所限省略了详细的分析过程，具体分析详见《下一代安全技术研究报告》[1]），进而归纳推导出下一代安全的主要发展趋势及研究方向——安全运营、安全智能、云及虚拟化安全、数据安全以及 CII 安全等 [1]。

3.1 安全运营

安全运营是研究安全产品互联的体系化防护及安全运维服务新模式时引入的一个概念，其定义如下：

安全运营：维持“闭环”系统正常运行和持续改进的周期性行为的总和，以“安全态势”信息为增速剂，按照指定响应时间的度量标准确保交付质量，最终促进客户环境、

生产环境和流转环境的持续改善。

通过开放的运营管理平台实现安全产品的互联，通过全局的安全情报采集与智能分析洞察所管理网络范围内的安全态势，并提供可视化的直观展示，进而基于安全态势结合最佳实践与专家知识进行快速的智能决策，并通过安全策略的快速分发、安全产品规则的智能配置以及基于云的安全服务模式，实现网络安全防护体系的快速响应。而安全情报采集、智能分析、评估与决策、快速响应服务组成的安全闭环持续、周期地运行，将能够实现用户网络安全环境的持续改善。

云安全服务体系、安全产品互联、智能安全配置管理、大数据管理与智能分析等将是安全运营平台需要解决的关键问题或技术，也将成为“下一代安全”研究的关注重点。

3.2 安全智能

安全智能主要是指智能信息处理及人工智能技术在信息安全领域的应用。

在现有的网络安全产品中，虽然已采用了一些智能信息处理技术，但多限于一些

基本的基于规则或策略的相关处理，诸如，SIMS 的报警关联、安全检测类产品（IDS、IPS、AV 等）与安全评估类产品（漏洞扫描）基于模式匹配的检测评估技术等。安全产品的分散部署、独立管理的现状使得各种安全信息难以得到共享和综合应用，缺少足够规模的高质量安全数据是当前网络安全领域走向更进一步智能化的一个关键瓶颈 [6]。

安全运营概念的提出，为安全智能技术提供了更大的发展空间。开放的安全运营平台可以汇集来自互联的各种安全产品的安全数据（日志、报警等）、监测信息以及利用蜜罐、蜜网或其他方式获得的安全威胁信息等，从而解决安全分析数据不足的问题。而基于大规模安全数据的大数据管理、入侵行为模式挖掘分析、全局网络安全态势评估、安全信誉评估 [2]、威胁情报分析以及各种自动化配置管理工具开发等安全运营相关的工作，都将离不开智能信息处理技术以及人工智能技术的支持。

综合上述，除 SIMS 的报警关联、IDS/IPS 的模式匹配检测、异常流量的检测与清

洗之外, 安全信誉 [2][7]、安全态势感知、威胁评估、安全度量 [8]、行为异常检测、自动化配置管理等 "安全智能" 的相关概念、理论和技术的发展及应用也将成为 "下一代安全" 研究的重要内容。

3.3 云及虚拟化安全

云计算及虚拟化安全的关注重点是云服务与虚拟化技术所带来的新安全问题以及如何利用云与虚拟化技术提高安全服务能力。主要涉及云计算平台的安全性、安全产品的虚拟化、安全服务云模式等几个方面。

其中, 云计算平台的安全性重点关注虚拟机安全以及云相关的软硬件、网络与应用协议的脆弱性。安全产品的虚拟化则关注使用虚拟化技术实现安全产品功能或安全产品的虚拟部署等。至于安全服务的云模式, 则主要考虑三种情况 [3]:

- 针对用户的安全云 (Security Cloud For User): 将云服务和安全功能绑定, 打包提供给用户, 比如 FireHost 推出的安全虚拟主机。

- 针对用户的云安全 (Security For Cloud User): 为使用各种云服务的用户提供

附加的安全解决方案。CipherCloud 的服务可以为用户使用的 Gmail、Amazon、Salesforce、Office365 等在线服务提供统一的内容加密功能。

- 针对云服务商的安全性 (Security For Cloud Vendor): 安全厂商的虚拟设备无缝地接入云服务商的环境中, 并作为可选插件提供给最终用户。VMware 的 Rob Randell 曾详细阐述了这一思路的优势和实现步骤 [5]。

随着云及虚拟化技术在 IT 应用领域的日益成熟, 也必将带来安全产品的部署与服务模式的变化, 安全运营平台的资源优化配置以及 SaaS 化安全服务就将是一种典型的应用。综合上述, 安全资源的虚拟化及云服务模式也将是 "下一代安全" 的重要研究方向之一。

3.4 数据安全

数据安全主要关注数据在其生成、存储、传输、使用以及销毁的整个数据生命周期中的机密性、完整性与可用性, 是信息安全防护体系中最为关键的基础防护措施。重要信息系统敏感数据的安全保护、互联网上个人

隐私信息的防泄露、舆情分析、内容过滤等都是当前数据安全研究的关注点。

云计算及虚拟化技术的发展造成了数据的存储和处理模式的改变: 大规模数据的安全存储与授权使用、用户数据在云中存储、处理时的可信性验证机制等, 这些变化将为数据的安全保护带来新的挑战; 同时, 移动互联网的普及、个人隐私数据的保护等也对数据安全提出了更高的要求。

由于我们专注的重点不是数据安全, 因此, 我们关于数据安全的讨论将局限于安全运营服务体系中安全信息的安全存储、授权使用以及安全产品中配置信息的授权访问等部分研究内容。

3.5 CII 安全

CII 安全主要研究作为国家关键信息基础设施 (CII) 重要组成部分的电力、交通、石油化工等行业信息系统的的核心安全问题。

近年来, 出于政治、军事、意识形态的目的, 敌对方的 "网络战" 威胁越来越频繁, 在这些攻击活动中, 国家关键信息基础设施、重要行业的工业控制系统、重要的信息系统都将会成为重要的攻击目标, 要

么破坏其可用性，要么窃取敏感数据，甚至是掌控舆论影响社会稳定。因此，加强 CII 安全防护极其重要，已成为“国家安全战略”的重要内容，也必然成为未来信息安全领域的研究热点与重要的业务增长点 [5]。

最近几年针对 CII 的攻击事件表明，这些攻击多采用有组织的、目的性很强的新型攻击手段（例如，高级可持续威胁——Advanced Persistent Threat，简称 APT）。为达成 APT 攻击，需要长时间地集中高端人才和技术，需要具备无孔不入的情报收集能力，往往掌握最新的 0-day 漏洞，拥有能够规避当时检测工具的传播和控制程序，以及利用所掌握资源快速展开连续行动的组织力和行动力。显然这样的攻击不是依靠单一技术就能防范和检测的，需要多层面安全措施的综合防御，对安全厂商及相关研究机构的安全服务能力提出了更高的挑战。

随着工业控制系统的智能化发展（比如智能电网、物联网等）以及互联网技术的应用，工业控制网络的封闭性也逐渐被打破，但工业控制系统因早期相对独立、采用专用协议而造成的安全性设计不足的问题，并未随着工业控制网络封闭性的打破而得到及时的解决。这些重要系统的固有缺陷，再加上敌对方的有目的攻击，使得 CII 系统，尤其是工业控制系统的安全必将成为“下一代安全”的研究热点，在业内也必将出现针对工业控制系统安全的“下一代安全产品”行业专版。

此外，智能终端和移动互联网的快速发展，基于无线接入的网络安全以及智能终端相关的安全问题也是当前安全研究的热点之一。

4. 结束语

综上所述，安全运营、安全智能、云及虚拟化安全、数据安全、CII 安全以及移动互联网安全等将是“下一代安全”的重要研究方向，并不可避免地影响“下一代安全产品”的功能特性和产品形态。

其中，安全智能、虚拟化、云服务及安全运营的相关技术与服务模式，将为安全厂商提供专业化的威胁情报分析及在线安全运维服务奠定基础：通过安全产品的互联，构建基于“闭环控制”的安全运营体系，建立安全信息（日志、报警、威胁情报等）的汇集、智能化分析与安全态势评估体系；实现针对安全威胁提供正确应对与及时响应的能力；并通过持续的安全运营实现用户网络环境的安全状态持续改善的目的。显然，在安全运营体系框架下前端安全产品的重要性将会下降，而云端服务的重要性则会得到提升 [6]。

参考文献

1. 李鸿培，下一代安全研究技术报告，2013.3
2. 李鸿培，信誉技术在安全领域中的应用，2011.5
3. 鲍旭华，多样性引发的安全知识变革 - 浅谈 RSA 2013
4. https://ae.rsaconference.com/US13/connect/sessionDetail.ww?SESSION_ID=1693
5. 李鸿培、于旻、忽朝俭、曹嘉，工业控制系统及其安全性研究报告，2012.12
6. 赵粮，下一代安全的思考 - 应对下一代威胁，NCI，2012.11
7. 卢小海，一种基于信誉的威胁分析方法，2010
8. 王卫东，安全度量综述，2010

从“软件定义网络” 到“软件定义安全”

行业技术部 王卫东

关键词：SDN SDDSec

摘要：本文前半部分简要介绍了“软件定义网络”（SDN）的概念和原理。考虑到安全网关类产品与网络交换设备在数据包处理方面具有相同的本质，可以将 SDN 的理念延伸到安全网关类设备的架构设计中，并将这种延伸称为“软件定义安全”（SDDSec）。后半部分阐述了 SDDSec 架构的组成与具体产品实现思路。

1. 前言

开放与封闭之间的较量，始终伴随着信息技术的发展。在主机领域，已经基本上完成了封闭系统向开放系统的嬗变，而网络设备领域，依然保持传统的体系架构和工作模式，这种架构和模式已经无法满足网管人员的“简单、高效、灵活、通用”等要求。越来越多的网络运维人员被下面这些问题所困扰：

- 设备如同铁疙瘩，很难集成、自动化排错 (trouble shooting)
- 不同厂商的设备之间互操作性不好，容易造成供应商锁定 (vendor locking)
- 厂商响应速度太慢，无法满足业务部门需求
- IP 地址和物理位置绑定，很难做业务迁移
- 传统路由策略太复杂，很难管理，容易失控
- 网络规模很难快速扩容收缩 (Scale Up/Down)

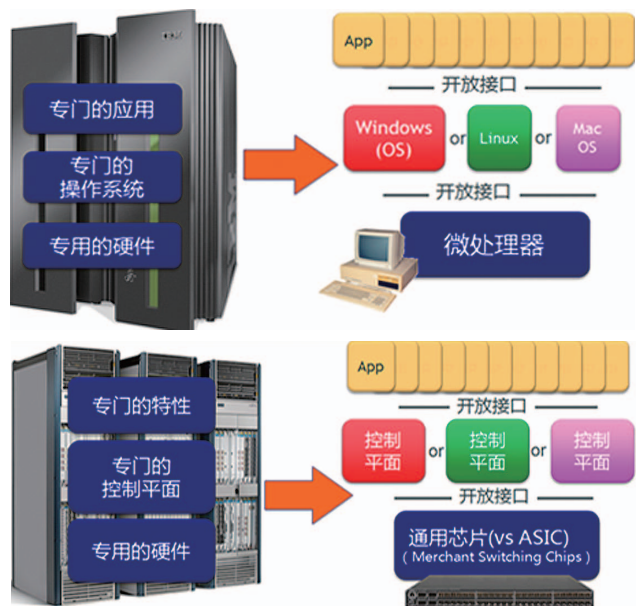


图1 PC 的出现对比 SDN 的发明 [1]

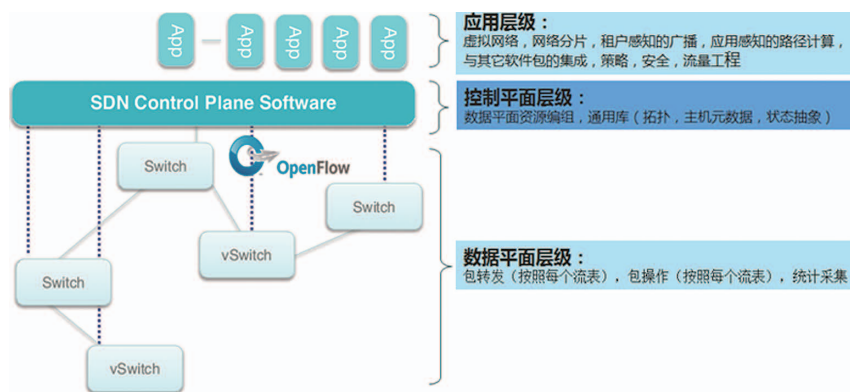


图2 SDN 的体系架构 [2]

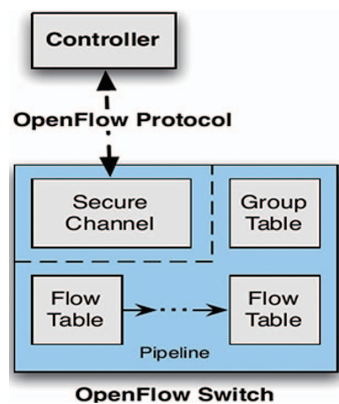


图3 OpenFlow 网络结构示意图

为了应对上述问题，网络技术即将迎来一次新的变革。这种变革的核心理念就是将网络设备的控制平面从设备中剥离出来，形成一种可以通过编程用软件定义的网络，简称SDN(Software Defined Network)。

如图1所示，SDN的发明与PC的出现在本质上是极其相似的，只不过一个发生在主机设备，一个发生在网络设备。SDN的体系架构如图2所示。

2.SDN 概述

2.1 SDN 定义和 OpenFlow 的工作原理

SDN是由美国斯坦福大学 Clean Slate 研究组提出的一种将控制从硬件中解耦合的新型网络架构，其核心技术 OpenFlow 通过将网络设备控制面与数据面分离开来，从而实现了网络流量的灵活控制，为核心网络及应用创新提供了良好的平台。简单来说，SDN可以实现网络如电脑般可编程，可以创建易于管理的网络虚拟化层，而 OpenFlow 则是一个标准化SDN应用协议。

OpenFlow 网络的组成

OpenFlow 网络由 OpenFlow 交换机和 Controller 组成。OpenFlow 交换机进行数据层的转发；Controller 对网络进行集中控制，实现控制层的功能。OpenFlow 网络的结构如图3所示。

- 控制器

OpenFlow 实现了数据平面 (Data Plane) 和控制平面 (Control Plane) 的分离，Controller 实现了控制平面的功能。Controller 通过 OpenFlow 协议这个标准接口对 OpenFlow 交换机中的流表进行控制，从而实现对整个网络进行集中控制。

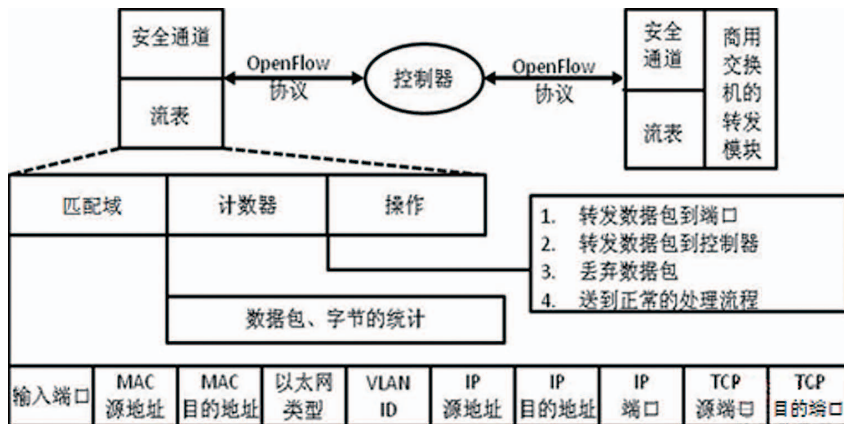


图4 流表的组成

● OpenFlow 交换机

根据白皮书的解释，一个 OpenFlow 交换机包含了最少三个部分：

(1) 流表 (Flow Table)

每个流表包含一组流项 (flow entry)，每个流项就是一个转发规则。每个流项包括匹配域 (match fields)、计数器 (counters) 和一组应用到匹配数据包的指令 (instructions) (如图4所示)，告诉交换机该如何处理这个数据流。

(2) 安全通道 (Secure Channel)

安全通道是连接 OpenFlow 交换机到控制器的接口，用于连接交换机与远程控制器 (Controller)，以便在控制器和交换机之间传递命令和数据包。控制器通过这个接口控制和管理交换机，同时控制器接收来自交换机的事件并向交换机发送数据包。交换机和控制器通过安全通道进行通信，而且所有的信息必须按照 OpenFlow 协议规定的格式来执行。

(3) OpenFlow 协议

OpenFlow 协议用来描述控制器和交换机之间交互所用信息的标准，以及控制器和交

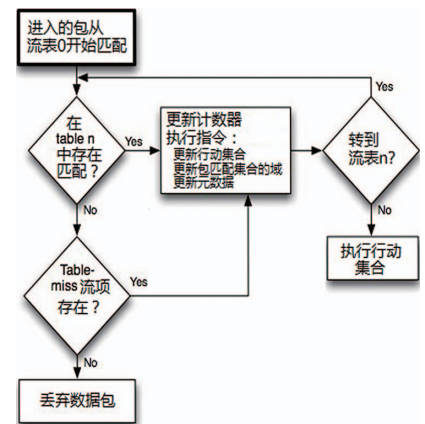


图5 OpenFlow 数据包处理流程

换机的接口标准。协议的核心部分是用于 OpenFlow 协议信息结构的集合。

● 数据包处理流程

如图5所示，当数据包进入交换机后，必须从流表0开始依次匹配。流表可以按次序从小到大越级跳转，但不能从某一流表向前跳转至编号更小的流表。当数据包成功匹配第一条规则后，将首先更新该规则对应的统计数据（如成功匹配数据包总数目和总字节数等），然后根据规则中的指令进行相应操作，比如跳转至后续某一流表继续处理、修改或者立即执行该数据包对应的动作集 (action set) 等。当数据包已经处于最

后一个流表时，其对应的动作集中的所有动作 (actions) 将被执行，包括转发至某一端口、修改数据包某一字段、丢弃数据包等。OpenFlow 规范中对目前所支持的指令和动作进行了完整详细的说明和定义。

2.2 OpenFlow 的用途

OpenFlow 可以被应用于很多场景，主要包括：

- 静态地划分 VLAN
- 定制路由协议 (比如单播、组播、多径、负载均衡)
- 家庭网络管理
- 移动性管理
- 能耗管理
- 流量控制
- 包处理器 (在控制器上)
- 访问控制
- 网络流量监控和虚拟化

其中 VLAN 隔离、访问控制和流量监控都属于网络安全的范畴。SDN 的内在性质就适合用于网络安全的场景。

3. 软件定义的安全

安全网关设备如防火墙 (FW)、入侵防护系统 (IPS)、抗拒绝服务系统 (ADS) 等同样存在通用网络设备的缺陷和不足，安全运维人员也同样面临前面所提到的困扰。例如通用的防护设备在用户那里使用效果并不理想，无法快速响应用户环境所特有的攻击，往往需要厂商的技术支持人员远程抓包和人工分析之后，找到一些攻击特征，再手工生成新的防护策略并部署到用户的防护设备上。有些文章称这种高度依赖人工分析且响应速度极慢的情形为信息安全的石器时代。此外，安全设备也同样存在互操作性差、难以集成等问题。造成这些困扰的根源在于安全设备的数据平面和控制平面是整合在一个设备上的。

所有的安全网关设备在体系架构和部署维护等方面与通用网络设备没有本质的差别，只是数据包转发规则是依据安全需求制定的而已，可以看作一类特殊用途的网络设备。基于这样的前提，SDN 的理念完全可以拓展到安全网关设备上，作为下一代安全产品开发的指导思想。即可以从 SDN 引申出 SDSec(软件定义的安全) 的概念，也

就是将安全策略执行点 (数据平面) 与安全控制器 (控制平面) 相互分离，从而使安全设备能快速、自动地适应业务的动态变化。SDSec 概念的具体实现表现在三个环节上，即海量数据的采集、智能和情境感知 (context-aware) 的异常检测和以流表项表达的安全策略。

3.1 海量数据的采集

在传统的网络中，数据采集可以利用串联在链路中的设备或分光镜像技术。但是在绝大多数异常检测的场合，只需要分析日志类型的信息，如数据包头信息 (也可看作流量日志)、操作系统日志、应用系统日志等等。

SDN 提供了一种新的数据采集方式。在 SDN 架构中，允许将指定的数据包发送到控制器。那些没有匹配到任何规则的数据包都可以看作是疑似异常的数据，有必要发送到控制器做进一步分析。对于那些匹配上的数据包，流表项中的计数器会记录相应的包数和字节数。

在 DDoS 攻击防护过程中，需要将攻击目标的流量牵引到防护设备上，这个牵引过程也可以看作一种数据采集过程。传统

▶▶ 专家视角

的牵引需要根据网络拓扑结构，选择用于牵引的网络协议，并在路由交换设备上做比较复杂的配置。而 SDN 的转发机制可以将牵引过程变得非常简便和快捷。

3.2 智能和情境感知 (context-aware) 的异常检测

在复杂的网络中发现入侵行为已经证明是很困难的。隐蔽的攻击者已经突破了传统的防御措施，如入侵防御系统、防火墙和杀毒软件，“大数据分析 (BDA)” 的方法将提供新的希望。有分析师在 RSA 关于大数据和大数据如何帮助增强安全的分析师小组讨论会上说：“统计分析将识别出异常情况，但是统计分析不理解安全。”他还预计最终将出现一个大数据的“安全算法市场”。已经有商业公司正在致力于利用大数据计算分析发现异常情况的研究。

传统的安全网关，异常检测和数据包转发处理往往集成在一个设备中。SDSec 的架构使得异常检测（相当于网络设备的控制平面）从安全网关中剥离出来，安全网关只负责根据流表进行数据包的转发处理。这样做有利于厂商集中精力研发异常检测技术，提高安全产品的核心能力，同时也便于在一个计算能力更强的、开放的、综合的平台上完成异常检测，例如将数据集中到云计算平台上进行异常检测的计算分析。

传统的异常检测往往是基于特征的静态对比，没有考虑时间因素。在实际环境中，同样的访问行为在不同的时机条件下，实际的效果不同，最终被判断的结果也应该不同。基于大数据分析的异常检测可以综合时间因素对访问行为做出判断。

在一个开放的平台上利用统计算法对多种来源数据进行分析计

算，并结合查询漏洞库、特征库、信誉库等各种信息库的结果，做出综合判断。这种检测过程较传统的特征比对的方法更容易获得准确的检测结果。

3.3 以流表项表达安全策略

安全策略一般是由“规则 (Rule)”和“动作 (Action)”组成的，而 OpenFlow 的流表项也有同样的组成元素。因此流表项可以很适合作用来表达一个安全策略，例如防火墙的 ACL 完全可以写成流表项的形式（如图 6 所示）。

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	*	*	*	*	22	drop

图 6 流表项形式的防火墙 ACL

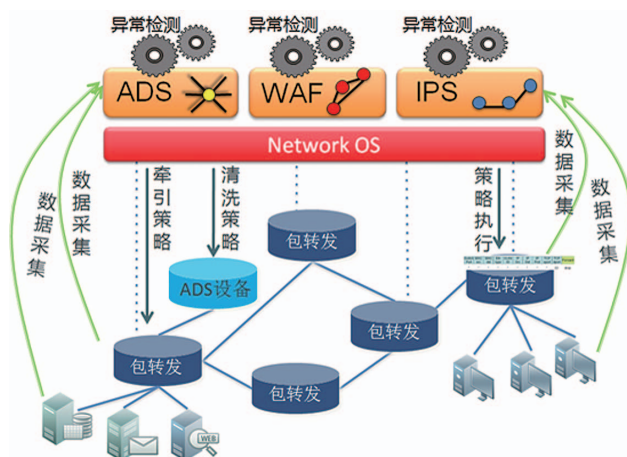


图 7 SDDSec 设备的体系架构和实现

在 SDN 架构下，传统的安全设备将分裂为两个部分：一部分是前面讲的异常检测，这部分具有高度的分析智能；另一部分是负责数据包处理的硬件设备。数据包处理的依据就是表达安全策略的流表项，它是由通用芯片（相对与 ASIC 而言）构成的低智能高性能的设备。分析平台负责将检测结果生成流表项，并下发到硬件设备上。

3.4 基于 SDSec 的设备体系架构和实现

攻击防护过程通常可以分解为数据采集、异常检测、策略执行三个环节。如图 7，基于 SDSec 的设备依然遵循这样一种体系架构。异常检测引擎或平台对采集来的各种数据进行分析，依据分析出的异常结果生成安全策略，最后安全策略下发到网络设备或安全网关上执行。

在性能允许的情况下，安全策略可以作为专门的流表下发到网络交换设备上；反之，安全策略需要在专用的安全网关上执行。对于串联部署的情况，安全策略全部在安全网关上执行；而旁路部署的情况，需要将待处理的流量牵引到安全网关上，流量牵引策略是在交换设备上执行，安全防护策略在专

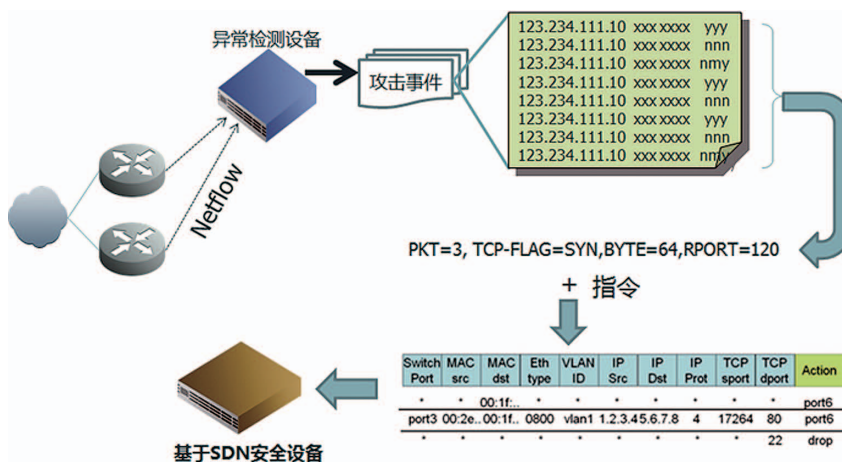


图 8 基于 SDSec 的 ADS 产品设计思路

用设备上执行。例如图 7 中，流量清洗策略在 ADS 设备上执行，而牵引策略在包交换设备上执行。

实现基于 SDSec 的防火墙，就是将防火墙中的 ACL 转换为一张 OpenFlow 流表，可以分布在各个网络交换设备上，也可以集中在一个专门的硬件上。FloodLight 项目已经开发出防火墙应用作为该项目的一个模块，允许在基于 OpenFlow 的交换机上执行 ACL (Access Control List) 规则。

图 8 给出基于 SDSec 实现 ADS 产品的设计思路。在大流量环境下，DDoS 攻击检测通常采用对流量日志（如 Netflow、sFlow 记录）统计分析的方法。异常检测设备检测到攻击事件后，可以很容易在原始数据中找到与攻击相关的流量日志。从攻击流量生成的日志中可以提取攻击特征。把攻击特征和 SDN 规范中的指令组合在一起，就构成了一个流表项。将流表项下发到基于 SDN 的安全设备上，凡是匹配上该流表项的流量一律丢弃，从而实现攻击流量的清洗。

4. 结束语

SDN 的开放架构具备很多优秀的特性，这些特性很好地满足了业务快速变化的需求，帮助用户摆脱传统专用硬件设备带来的很多困扰。这些特性对于网络安全的应用场景中尤为重要。而 SDSec 作为 SDN 的在安全设备上的应用，全面继承了这些特性。

4.1 简易性

基于 SDSec 的安全设备，通过集中化的控制和流表化的安全策略，可以大大简化安全设备的部署和维护。

4.2 高效性

SDSec 架构允许快速动态的部署安全策略。从控制器向安全设备下发策略比传统的登录到设备上手工配置策略要快捷很多。用户可以根据自身的需要，以软件开发的速度创新和定制安全策略，而不必依赖来自厂商响应缓慢的更新升级和技术支持。

4.3 可伸缩性

在 SDSec 架构下，安全设备可能演化成交换设备上一个流表（或流表项），安全设备的部署和撤除演变为流表（或流表项）的启用和关闭，这些都可以通过控制器上的简单操作来完成。

4.4 互操作性

基于 SDSec 的安全设备具有高度的通用性，不同厂商的设备可以很容易互相替代，用户可以避免“供应商锁定”的困境。安全设备的核心能力聚焦在异常分析，并以各种 APP 的形式存在。用户可以与多个合作伙伴或厂商交换技术。

4.5 创新性

开放的控制平面管理接口，网管人员能够使用网络 API 来编写软件，按本地需求定制安全策略，消除不需要的设备内建策略。

业界对 SDN 的看法尚存争议，也会同样影响对 SDSec 的评价。有些人认为 SDN 并不是什么新概念，未来在产业化过程中还需要克服很多技术困难，例如性能与硬件成本和功耗之间的矛盾、架构设计很难兼顾性能和效率两方面的要求。更有行业分析机构质疑：为了引入可编程性使用复杂的且可能令网络不稳定的代码值得吗？

另一种观点对 SDN 的前途充满信心。产业内大厂商的一系列收购行为（Oracle 收购 Xsigo、VMware 收购 Nicira、Juniper 收购 Contrail、博科收购 Vyatta、思科收购 Cariden 等），可以看作新技术被接受的标志之一，因为被收购公司的技术至少是值得追求的，而且比起自己开发，更好的方式是购买。一些主流网络设备厂商推出基于 SDN 的设备，也标志着这项新技术被业界广泛接受。

第三种观点认为 SDN 技术用于交换领域是个错误的方向，而最适合的应用方向应该是在安全领域。甚至有专家认为，SDSec 将颠覆现有的 IT 安全模型。

参考文献

[1] Nick McKeown, "How SDN will shape networking"

[2] Dan Pitt, "SDN Standards : What and Whatnot!"

[3] "OpenFlow 网络的基本组成" <http://network.51cto.com/art/201105/264166.htm>

数据中心Web系统 协同防护思路探讨

产品推广部 田民

关键词：DDoS 防护 数据中心 Web 安全

摘要：数据中心里的 Web 网站系统面临的网络安全威胁包括应用层攻击、网络层攻击和混合攻击。传统被动、单点以及彼此孤立的防护手段已不能应对越来越严峻的安全威胁。改被动防御为主动防御，变单点防御为分层防御，变孤立的防御为协同防御，配以 24x7 全天候的安全监控和应急响应，实现快、主动和准确的防护目标。

一、引言

无论是个人还是企业，对互联网的依赖在不断增强。在商务应用中，Web 技术正在承载着越来越多的核心业务。但不幸的是，目前 75% 以上的网络攻击都瞄准了网站系统 (Web system)。技术上讲，网站是一套复杂的系统，在包括云计算等新兴技术的驱动下，功能的复杂度还在不断增加。对网站的安全防护，作为一种属性或能力，却远远地落在了新技术和新功能的后面。

这个问题在数据中心市场显得尤为明显。托管在数据中心中的 Web 系统对于企业来说，是联系企业及其客户的桥梁。然而，绝大部分安全专家认为，目前数据中心中的 Web 应用存在很大的安全挑战，当前的安全措施普遍落后于日新月异的攻击方法。

要保护数据中心 Web 应用系统的安全，首先必须对网站威胁进行深入的研究。从网络入侵的角度来看，网站面临的威胁可以分为三大类：

- 1) 应用层攻击
- 2) 网络层攻击
- 3) 混合攻击

二、威胁

2.1 应用层攻击

漏洞是网站系统面临的最大的威胁。入侵者对网站系统的入侵很大程度上依赖于网站系统存在的漏洞。可以说，Web 应用安全问题本质上源于软件质量问题，质量问题

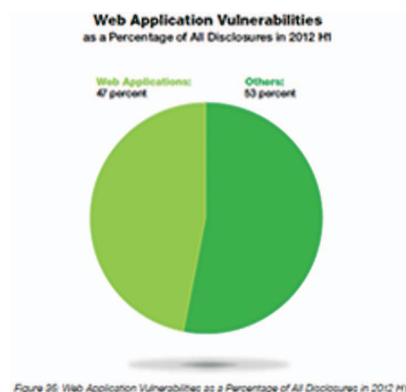


图1 Web应用漏洞所占比例

来源：

X-Force 2012 Mid-year Trend and Risk Report, IBM

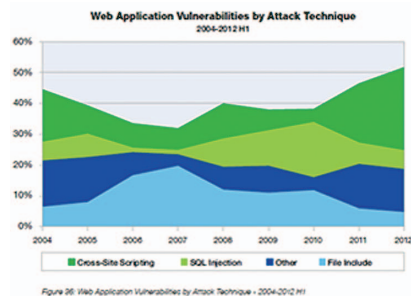


图2 Web应用漏洞与攻击手段的对应

来源：

X-Force 2012 Mid-year Trend and Risk Report, IBM

的直接体现就是漏洞。漏洞可以分为系统漏洞和应用漏洞。总体来说，应用漏洞的增长趋势是非常明显的。

根据 IBM X-Force 2012 Mid-year Trend and Risk Report, Web 应用漏洞从 2011 年的 41% 增长到 2012 年上半年的 47%，如图 1 所示。

由于 Web 应用往往是某个机构或组织所独有的应用，对其存在的漏洞，发现和防护总出现滞后性。这也是为什么总有 0-day 漏洞不断被利用的原因。已知的漏洞缺乏时效性和有效性，利用签名和特征库很难满足漏洞防护的需要。

同样源于 X-Force 2012 Mid-year Trend and Risk Report, Cross-Site scripting 和 SQL Injection 是 Web 系统最大的应用层威胁，如图 2 所示。这和 OWASP 2010 版 TOP10 的结论是一致的。

2013 年 3 月美国 NIST 维护的国家漏洞库 (NVD) 遭到入侵的原因就在于入侵者对 Adobe ColdFusion 软件漏洞的利用。入侵者通过利用 Adobe ColdFusion 软件漏洞获得了服务器访问权限，在服务器上安装了恶意软件，下发攻击其他服务器的命令。

对于漏洞的防护关键是快，越早发现漏洞并弥补，被入侵者利用的概率就越小。

2.2 网络层攻击

网络层威胁，从攻击的角度上，普遍被称为带宽型攻击。DDoS 是带宽型攻击的一种典型方式，也被称之为流量洪水攻击。在数据中心的环境里，通过发起大流量的 DDoS 攻击来阻塞出口带宽是常见的一种攻击手段。

DDoS 攻击通过发出海量数据包，造成设备负载过高，最终导致网络带宽或是设备资源耗尽。通常，被攻击的路由器、服务器和防火墙的处理资源都是有限的，攻击负载之下它们就无法处理正常的合法访问，导致服务拒绝。DDoS 攻击把大量看似合法的 TCP、UDP、ICMP 包发送至目标主机。

根据 NSFOCUS Security Threat Report 2012, DDoS 攻击仍旧是 Web 系统最大的

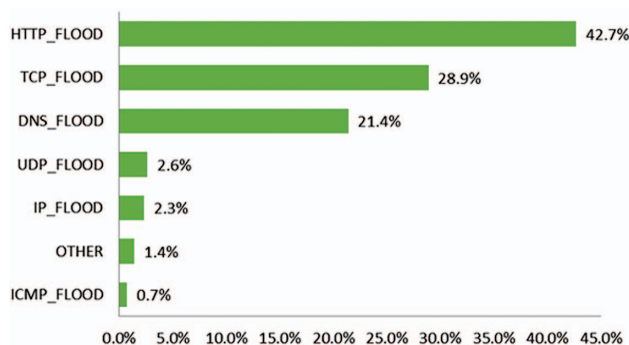


图 3 DDoS 攻击组成

来源：Security Threat Report 2012, NSFOCUS

威胁之一，如图 3 所示。其中，HTTP Flood、TCP Flood 和 DNS Flood 分列前三位。

网络层攻击产生的流量是非常惊人的。从 2013 年 3 月 18 日起，欧洲一家名为 Spamhaus 的公司开始遭受 DDoS 攻击。攻击者通过僵尸网络和 DNS 反射技术进行攻击，攻击流量从 10G 不断增长，在 3 月 27 日达到惊人的 300G 攻击流量，被认为是互联网史上最大规模的 DDoS 攻击事件。

2.3 混合攻击

在 NSFOCUS Security Threat Report 2012 中提到，当前针对 Web 系统的攻击威胁还体现为一个重要趋势——混合攻击，即将应用层攻击和网络层攻击混合起来，并利用了 Web 系统的种种漏洞。混合攻击使得任何单一层面的防护手段失效。举一个实际的例子，绿盟科技在对 Anonymous 组织若干攻击行为的分析后发现，Anonymous 攻击就是典型的混合攻击，DDoS 只是他们最引人注意的方面罢了，如图 4 所示。

防护混合攻击对于安全运维团队的要求是非常高的。当前很多混合攻击呈现 APT 的特点，入侵者往往会利用 DDoS 攻击作为诱饵吸引防护一方响应团队的注意力，而其主要的目的在信息窃取等更深层面。因此，运维团队的技能、经验和效率都是能否有效防护进攻的重要因素。



图 4 Anonymous 攻击分析

来源：Anonymous Report for Operation Japan, NSFOCUS, 2012

三、挑战

从上文的分析中，了解了数据中心环境下，Web 应用系统所面

面临的威胁是非常严峻的。不管攻击的一方是采用单一形式的攻击，还是采用混合多种手段的混合攻击，作为防护一方，你需要考虑如下几个问题：

- 1) 在攻击发起前，能否先于攻击者发现系统存在的漏洞？
- 2) 在攻击过程中，能否快速地发现、响应和控制？
- 3) 安全运维团队的能力和效率怎样？

决定上述问题的关键是一——时间。攻防双方，哪一个占据了时间上的优势和主动，哪一方将获得最终的成功。

3.1 是否以更为主动式的防御来取代完全被动的防御？

如果你可以在攻击者发起攻击之前，采取主动式的防御机制，先于攻击者发现系统的漏洞并有效修复，那么无疑减小了系统的攻击面，降低了系统面临的风险，从而客观上延后了系统可能遭到攻击的时间。然而，如果你的防御仍旧是静态和被动的，不能实现事先的检查和防护，那么等于将战场的主动权交给了你的对手——攻击者，什么时候开始这场攻防战，你的对手说了算。

3.2 安全防护设备是否能够对攻击行为进行针对性的防护以及彼此配合？

如果对攻击的发现来自于 Web 服务器资源枯竭的告警，甚至是客户的投诉，那么就太晚了。特别对于一些提供在线交易的 Web 系统，业务的可获得性和连续性至关重要，攻击需要被尽早发现和及时处理。这些挑战最终需要安全设备来解决。

首先，不同类型的安全防护设备的检测和控制原理是不同的，这是由攻击者攻击手段的不同决定的，因此，不要指望在一个设备

上同时实现对应用层攻击和网络层攻击的防护。

其次，是采用在线式还是旁路式来进行防护，对于不同的攻击，取得的效果可能就存在差别。在线式防御的响应速度快，对于应用层攻击的防护效果好。

最后，在面临诸如混合攻击的复杂攻击情况下，设备间的协同配合就显得非常重要。如何实现快速调度不同设备的防护能力，是解决复杂攻击的关键。自动化程度越高，响应时间越短，效率越高。反之，如果设备间的协同和调度仍通过人工来进行，那么你的响应时间可能是几个小时。

3.3 安全运维团队是否处于 24x7 的工作状态以及是否足以应对最严酷环境下的攻防对抗？

攻击者的时间表是 24x7，不管是深夜，还是节假日，他们随时都可能发起进攻。这样一来，维护人员能否在攻击发生后的第一时间到现场进行对抗性操作直接影响到防护的成败。同时，你的维护团队是否真的可以熟练操作和调度安全设备来完成严酷攻防对抗场景下的工作也是一个决定成败的重要因子。坏消息是，不要指望一般的系统维护人员同时具备安全攻击的应对能力，事实上，专业的安全专家总是最稀缺的资源。如果你的运维团队不能在第一时间发现攻击并熟练操作和调度安全设备，直接的后果就是你对攻击的响应时间可能又是几个小时。因此，你需要的是一支专业化和全天候的安全应急团队。

四、解决方案

4.1 目标

毫无疑问，对 Web 系统的防护是一场与攻击者进行的赛跑。

对于攻击的一方来说，如果先于防护一方发现 Web 系统中存在的漏洞，就可以用最短和最高效的方式入侵 Web 系统，获取数据。此外，时间因素在黑客的攻击过程中同样重要。在防护一方采取一定的防护措施后，攻击一方越早发现攻击受阻并转换攻击手段，就可以获得更高的攻击成功率。

转换视角，对于防护一方来说，时间因素同样决定了防护的成败。

首先，对于防护的一方来看，如果先于攻击一方发现 Web 系统中存在的漏洞，尽早修复它们，就可以防患于未然，获得最低的防护成本。漏洞的修复方式并不是一定要依靠修改网页代码才可以实现，对于一些暂时不能修复或需要投入较长时间才能修复的漏洞，可以通过部署安全设备和相应的规则进行防护。通过修复漏洞获得防护能力的提升，可以实现以最小的成本获取最高的防护效果，因此，漏洞修复是网站防护优化的重要工作。

其次，设备的防护效率也是一个非常重

要的因素。特别对于混合式攻击来说，不同层面的攻击需要不同的防护思路，对应到不同的防护手段 / 设备上。对于多层面攻击的防护，安全设备提供的防护面应该是无缝的。不同安全设备不应是孤立的，需要彼此配合。设备协同响应的速度越快，配合的效率越高，防护效果越好。

最后，在攻击发生后，安全人员能否在最短的时间发现攻击并做出响应是决定防御成败的关键。越早发现攻击，留给响应处置的时间就充裕。同时，响应的速度越快，部署有效的控制手段的时间就越短，系统受到的损失就越小。从发现到响应再到控制，是一个递进的关系。从应急响应流程上来说，任何一个环节效率的丧失都直接影响防御的效果。

因此，尽可能地将防护时间提前，提高检测和防护的效率，缩短响应的时间，对于防护一方来说，是解决时间这一关键问题的所在。

4.2 思路

考虑到数据中心环境下的网络和应用特点，最终要解决的问题是应对 Web 系统面

对的威胁。从整体防护方案的设计上讲，主要体现了五个方面的思考。

1) 主动防御的思路。对威胁的防御不仅要考虑到攻击过程中，还要考虑到事前的防护。尽早发现系统中存在的漏洞，争取更多的时间来进行系统的修复，实现更为主动的防御；

2) 分层防护的思路。对于大流量多类型的攻击需要考虑在不同层面进行有针对性的防护。在流量的上游（数据中心出口）部署对大流量网络攻击的防护设备实现网络层攻击的防护，在下游 / 客户端过部署 CPE 防护设备处理小流量的应用层攻击。如何做到因地制宜，最大程度上发挥 CPE 以及上游流量型安全防护设备的能力是考验系统整体防护设计和效率的重要方面。

3) 协同防护的思路。对应用层攻击的防护与网络层攻击的防护手段不是割裂的。攻击流量在不同防护设备之间进行调度，以实现应用层攻击防护设备与流量型防护设备之间的协同工作，是至关重要的。

4) 知识共享机制。跨用户以及跨地域的安全防护知识共享可以很大程度上提升防护

的效率，取得事半功倍的结果。

5) 运维的考虑。安全运维依托人来完成，而安全攻防最终是人之间的对抗。人的因素不可忽视。设备是由人来操作的，不同的人对设备操作的结果是不同的。操作设备人的经验、技能将直接影响到防护效果。

4.3 设计实现

首先，对于应用层攻击的防护，需要解决漏洞的问题。通过远程对 Web 系统进行扫描，发现系统中的 Web 漏洞。在一般的防护实现中，扫描的结果一般就是报告的形式。我们知道，如果通过修改网站代码来修复这些漏洞是非常耗时和费事的。如果可以将扫描的结果导入防护设备中，生成防护规则，基于这些“虚拟补丁”实现对漏洞的修复以及系统的防护，无疑可以大幅提高防护的效率。

其次，当入侵者发起攻击之后，就需要基于签名/特征以及会话重组发现攻击行为，进行数据包的过滤。对于 Web 系统来说，应用层攻击主要包括诸如 SQL 注入、跨站脚本等攻击。应用层设备部署在客户现场，即 CPE 端设备。通过在线式的防护，实现

最快速的响应和最精准的防护。对于网络层攻击的防护来说，攻击的发现基于对特征/签名以及流量行为的检测。流量型防护设备一般来说都是旁路的，离线式的清洗中心是最常见的部署方式。部署在客户侧和清洗中心中的设备分别负责应用层防护和流量型防护，是分层防护思路最典型的体现。

需要特别说明的是，应用层防护和网络层防护不应该是彼此孤立的。我们知道，入侵者在攻击时，往往会采用混合攻击的方式，既有类似 TCP/UDP Flood，也有诸如 HTTPS Flood 这样的攻击。客户端设备往往因性能的限制无法处理这样的混合攻击。在这样的情况下，需要进行攻击流量防护工作的调度，即一旦部署在客户端的防护设备发现攻击流量超过了自己可以处理的能力，需要立即通知上游的清洗中心流量型防护设备，将攻击流量的清洗工作交给流量型清洗的设备来进行。这就是协同防护思路的体现。

第三，防护一方需要实现信息共享。攻击者在论坛和聊天室里讨论、沟通和分享攻击的经验，使得攻击手段和工具在不同的区域甚至国家间快速传播。如果防护的一方采

用孤立无援的防御策略，对于攻防双方来说，就太不公平了。你可能不知道，面对的攻击可能其他同行已经领教过，并已成功防护了。如果你能够在最短的时间内获得来自其他已经成功防护此类攻击的同行的防御策略，你的成功可能简单到只是点击几下鼠标按键。事实上，安全信息共享的重要性正是 Izz ad-Dinal-Qassam Cyber Fighters 对美国银行的 DDoS 攻击案中带给安全业界的宝贵经验。美国货币监理署办公室(OCC)建议银行间实现信息共享，建立风险缓解策略的共享。这就是建立防护知识共享机制的体现。

最后，出于对运维的考虑，支撑上述防护系统有效运行的是一个高效运作的专业化安全应急团队，保证上述所有安全防护思路和机制的快速和有效的落实。由于攻击者的时间表是 24x7 的，因此这个应急团队的工作也应该是全天候的。

五、总结

综上，一个全天候的主动、分层和协同防护 Web 安全方案建立在快速发现、快速

响应和快速防护的基础上。具体来说,通过对 Web 系统的检测将防护提前到攻击发起之前,并通过分层防护保证应用层和流量型攻击的有效处理。防护的调度及设备间的协同配合,可以将攻击的响应速度提升到分钟 / 秒级。同时, 24x7 专业的安全专家团队,从“人”的因素上,将响应时间从“小时”甚至“天”降低到“分钟”级。最后,跨区域的防护知识共享机制极大地提升了防护的效率。因此,方案设计的最终目标就是在最短的时间内完成防护能力的部署,以及快速有效地阻断攻击。

对于数据中心用户来说,方案可以化被动防御为主动防御,防患于未然,降低防护的成本。同时,分层防护以及设备的协同配合可以降低安全响应的时间,提升防护的效率。

对于旨在进行自身防护的企业或数据中心来说,将一部分安全的 Capex 转化为 Opex,通过服务外包的形式解决安全专家缺失带来的防护隐患。而对于计划开展安全防护业务 MSSP 来说,这种服务外包的形式可以保证业务的快速上线。

参考文献

- [1] X-Force 2012 Mid-Year Trend and Risk Report, September 2012, IBM
- [2] Security Threat Report 2012, 2013, NSFOCUS
- [3] Anonymous Report for Operation Japan, 2012, NSFOCUS
- [4] Matthew Shaer, Spamhaus targeted by most powerful DDoS strike in history,
<http://www.csmonitor.com/Innovation/2013/0327/Spamhaus-targeted-by-most-powerful-DDoS-strike-in-history>
- [5] Eric Chabrow, Vulnerability Floors Vulnerability Database Site
<http://www.govinfosecurity.com/vulnerability-floors-vulnerability-database-site-a-5615>
- [6] Tracy Kitten, Regulator Warns of DDoS, Fraud Link
<http://www.bankinfosecurity.com/regulator-warns-ddos-fraud-link-a-5379>

网络结构安全在电子政务中的应用

——两种典型网络结构的安全对比分析

行业技术部 张智南

关键词：电子政务 结构安全 等级保护

摘要：根据业务需求和安全设计思想的不同，各政府部门网络结构安全设计呈现出不同的形式。本文从典型部门政务网络架构出发，介绍了两种常见网络结构安全设计，对比分析了它们的设计思想和主要优缺点。

引言

在信息安全技术体系中，网络安全是一个重要组成部分，而结构安全又是网络安全的基础。安全的网络结构是实现信息系统安全的重要保证。本文通过讨论两种常见的基于安全理念设计的部门政务网络结构，对比分析其设计思想和主要优缺点，为安全体系架构总体设计提供一些可行的思路。

一、典型部门政务网络架构

从 1999 年政府上网工程算起，我国电子政务建设已经走过了十几年的历程。国家各级政府部门（特别是各部委级单位）作为电子政务网络建设的领头羊和主力军，以国家电子政务建设有关规定为

指导，结合本部门相关业务需求，逐步建立了对外连接互联网、对内部省市县纵向贯通、同时与协作单位横向互联的全国性部门政务网络。

遵照公众服务、业务发展和保守国家秘密等主要需求，各类政府部门在网络建设过程中基本形成了典型的外网、专网、内网三层网络架构，如图 1 所示。

外网，也称外部服务网、公共服务网，主要部署政府网站、外部邮件系统以及直接通过互联网服务公众的应用系统前台，是各级政府部门对外服务的窗口。专网，也称业务网、生产网，主要部署各类政府部门内部的核心业务系统和核心数据库，是业务数据集中处理和业务工作集中开展的平台。内网，也称涉密网、办公网，主

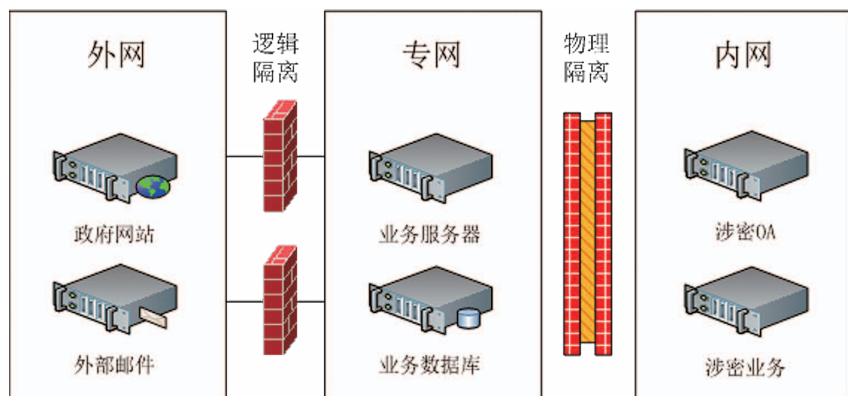


图1 典型三层网络结构

要部署各类政府部门涉及国家秘密的业务系统和办公系统，是处理部门内部与国家秘密有关信息的专用网络。外网网站、业务系统与专网业务系统、数据库往往是前后台关系。由专网业务系统为外网网站、外网业务系统提供数据支撑，外网与专网之间多数采取逻辑隔离方式。与此不同，内网重点考虑保守国家秘密的需求，因此与外网、专网之间通常采取物理隔离方式。

不同子网承载的业务不同，其外联需求也不尽相同。外网直接对公众提供服务，对外主要与互联网相连。专网作为承载核心业务和核心数据的骨干网络，一方面为满足业务系统通信、数据交换的需要，与本部门上下级单位纵向贯通，另一方面为满足与协作单位业务协同的需要，与协作单位横向互联。内网出于保密的需求，很少与外部网络直接互联。由于内网与外网、专网物理隔离，且基本没有互联需求，下文讨论的网络结构安全不包括内网部分。

二、基本思想与相关要求

网络结构安全设计的基本思想是“网络纵深防御”，即通过对网络的结构化划分，为在

网络不同位置建立互为补充的安全防护机制奠定基础，从而形成从外到内多层次的网络防御体系。

等级保护制度作为保障国家信息安全的基本制度和各类政府部门开展安全建设工作的主要依据，也基于“网络纵深防御”思想提出了相关要求。例如，在《信息安全技术 信息系统安全等级保护基本要求》（GB/T 22239—2008）中，二级系统网络安全结构安全部分规定：“应根据各部门的工作职能、重要性和所涉及信息的重要程度等因素，划分不同的子网或网段，并按照方便管理和控制的原则为各子网、网段分配地址段”。三级系统进一步规定：“应避免将重要网段部署在网络边界处且直接连接外部信息系统，重要网段与其他网段之间采取可靠的技术隔离手段”。这些规定为各级政府部门进行网络结构设计提供了基本依据和实施准则。

三、两种常见网络结构

在政府部门网络中，由于专网主要承载核心业务系统和关键业务数据，一般将专网作为重要子网进行安全防护。在网络实际建

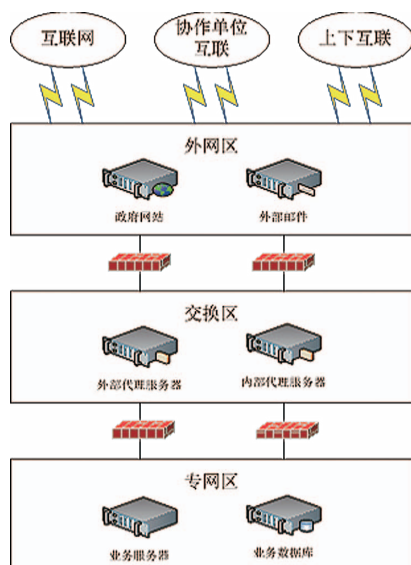


图2 “1+1+1” 模式

设过程中，根据不同部门的具体安全需求，以重点防护专网为中心，逐步形成了多种网络结构。总体上看，主要可以划分成两种结构模式，在此分别称之为“1+1+1”模式与“2+2”模式。

（一）“1+1+1” 模式

“1+1+1”模式如图2所示。

如图所示，“1+1+1”模式中，网络从外到内被划分为外网区、交换区、专网区三部分。

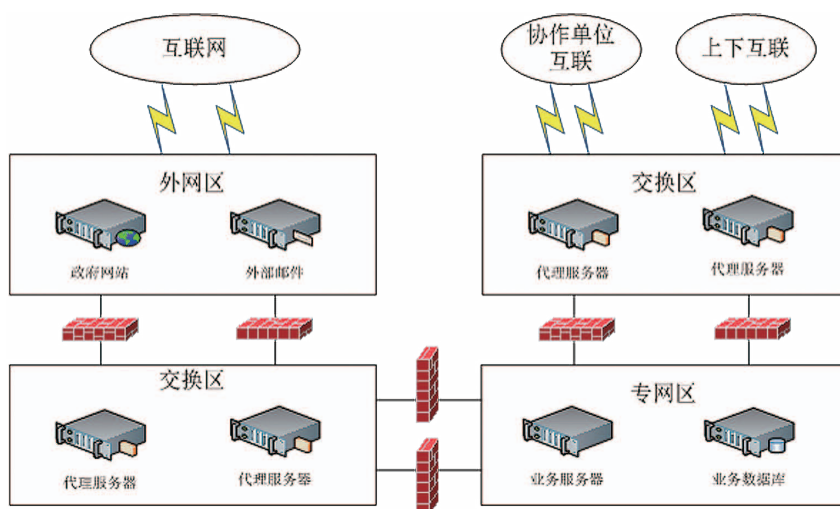


图3 “2+2” 模式

外网区主体上与前文所述外网部分重合，部署政府网站和外部邮件等系统。同时，外网区作为唯一与外部网络连接的子网，是网络唯一的对外接口。专网区主要部署部门内部的核心业务系统和生产数据库。在专网区与外网区之间设置交换区，采用落地交换方式完成外网区与专网区的数据交换。

（二）“2+2” 模式

“2+2”模式如图3所示。

如图所示，与“1+1+1”模式三个子网不同，“2+2”模式划分为四个子网。其中外网区不再是网络唯一的对外接口，仅提供互联网连接。协作单位互联及上下互联通过再设置一个独立交换区的方式实现与专网区的数据交换。

(三) 差异分析与优缺点比较

从合规角度，两种网络结构都满足等级保护的相关要求。其主要差异存在于对外部互联接口的设计。

“1+1+1”模式基于信息安全的“恶人假定”思想，认为所有来自外部的访问都是不可知、不安全的。同时从安全边界整合的角度出发，将所有对外出口统一整合到外网区，进行统一安全防护。

“2+2”模式对外部访问存在的威胁做了分类，对互联网访问与协作单位、上下级部门的访问区别处理。其中，互联网是完全不可信任的区域，来自互联网的访问需求中存在恶意攻击的可能性最大，因此采取最严格的访问控制、流量检测、入侵防御等安全机制进行防护。而协作单位、上下级部门属于来源基本可知、在一定程度上可信赖的区域，因此对协作单位互联、上下互联的安全防护强度可弱于互联网。

从安全的角度，两种模式各有其优缺点，如表 1 所示：

表 1 优缺点比较

网络结构	优点	缺点
“1+1+1”模式	1. 按照最高安全威胁进行统一边界安全防护，防护强度高。 2. 边界整合度高，网络结构相对简单。	1. 对边界安全设备性能要求高，投资相对较大。 2. 外网区多种接口并存，接口间隔离要求高。 3. 高强度安全防护与高实时性要求系统存在矛盾。

“2+2”模式	1. 高、低风险接口分离并采取不同安全防护措施，有效节约投资。 2. 对协作单位和上下级部门实时性要求高的业务系统支持较好。 3. 在不考虑互联网接入的基层部门，仅需对协作单位、上下互联接口进行安全设计，便于上下级部门边界安全策略的统一。	1. 对来自协作单位、上下部门的跳板攻击防护性差。 2. 多交换区设置，安全策略相对复杂。 3. 安全边界数量增加，可能的故障点也随之增加。
---------	---	--

四、结束语

安全建设的基本出发点是要满足业务运行的需要，网络结构安全设计也必须根据部门的实际情况与具体需求综合考虑。随着国家电子政务网络建设和应用工作的进一步开展，各政府部门专用业务信息网络将逐步与电子政务外网分别对接，形成一个纵横双向都很庞大的政务网络。在互联互通的大趋势下，不可避免地会引入新的安全风险，因此就更有必要做好各部门内部的网络结构安全。

参考文献

[1]GB/T 22239—2008 信息安全技术 信息系统安全等级保护基本要求

[2]发改高技[2012]1986号 关于进一步加强国家电子政务网络建设和应用工作的通知

水煮互联网——漫谈用户认证

行业技术部 罗爱国

关键词：账号 认证 加密

摘要：互联网飞速发展的今天，网络应用已经渗透到生活的方方面面。我们每个人都作为用户在众多的服务提供商处注册账号，以便享有差别化的服务。用户认证作为第一道也是最重要的一道防线，一直倍受关注。

引言

用户，亦称使用者，指使用产品或服务的自然人或法人（通常由指定的一人或多人负责）。

账号是一个较现代的说法，通常指代（标识）用户的某一角色。最常见的角色有公司里的 CEO、总监、经理、员工或者家里的儿子、丈夫、父亲……一个人通常拥有多种角色，在此，我们也可以说他（她）在多个系统里拥有账号。

认证是鉴别（核实）某人是某个角色的过程。怎么鉴别？这就涉及用于鉴别的凭据（证据，它是指在某种情形下，等同某人的替代物），以及怎样鉴别（认证）。是不是把简单的问题复杂化了？那就对了。

举一个例子。一天，你睡梦中接到一个电话，一人自称是你表弟，被抓赌的警察叔叔抓了个正着，急需 2000 元了结。你一听，有点蒙，印象中是有这么个表弟，可多年未联系，加以电话里噪音嘈杂，根本听不出来是谁，就准备撂电话。电话里头的人急了：“老表，那

次 xxx，还是我背的黑锅，这次你不能见死不救呀。”听到这里，你心里暗骂一句“死犊子，净不干人事”，嘴里却说“把卡号发过来”。

在这个例子里，用户是打电话的人，账号是表弟。这年头表哥表弟满天飞，表姐表妹一大把，到底是不是，确定不了。但关键的一条凭据证明了打电话的人就是你那个不争气的表弟，因为 xxx 的事知道的人本来就不多，而背黑锅就是你俩之间的秘密了。

在整个过程中，某人声明是谁没什么，关键是要能提供凭据证明他的声明。这其中，凭据是我们关注的焦点，也是一切问题的核心所在；其次，认证的过程也非常重要，它直接影响认证的有效性。接下来，我们将分别讨论凭据与认证过程。

一、凭据

说到凭据，除了常见的密码，你可能还想到了数字证书、USB key 等，不错，所有可用于鉴别账号（角色）的东西都可以作为凭据。不过在网络环境下，要求凭据可以通过网络传输，从而大大限制了可

使用的凭据种类。

一般来说，有三种凭据，即他知道的内容、他持有的物品、他就是这个人。

（一）他知道的内容

他知道的内容（可以理解为我们通常所说的信息）可以是密码、个人身份号码、母亲的乳名、或密码锁的密码。

金无怠并不认识这里的人，餐馆里的人也不认识金无怠，大家都是靠对暗号来确认身份的。暗号极为简单，就是按照一个特定的顺序点菜，金无怠坐下后就开始点，麻婆豆腐、清炒猪肝、排骨炖汤，然后在规定的这个菜后面问一个价格问题，给出一个数字。金无怠就问，排骨汤是2块3吗？若是对方心领神会的，立刻会给出另外一个数字。跑堂果然立刻说，先生，今天排骨汤打折，1块8毛5，然后，跑堂再推荐另外一个小菜，说，这里的杂碎做的不错，先生要一盘吗？金无怠知道对方是情报员，而此时客人只要点一下菜单上的这个地方，另外一只手抓三下头皮，对方就可确认了。金无怠就指着菜单上的一个菜名问，是这个李鸿章杂碎吗？边说边用另外的手抓了

三下头。跑堂立刻说，是的，先生点吗？金无怠说，要的。接下来，对方应该给送上一杯茶，说这是刚进口来的龙井，是新采的。金无怠等着，心里有些激动，因为这个历史上最重要的情报需要立刻传递出去，就从这里，让京城里知道。但是，在没有完成最后一个环节前，他不可以轻举妄动的。过了一会儿，店老板拿着一杯茶出来了，看见金无怠就笑嘻嘻地说，先生，请喝茶，这是刚进口来的龙井，今年新采的，您尝尝。金无怠知道这跑堂和老板都是情报人员，连忙道谢。

优点：通过他知道的内容进行认证是最经济便捷的。

缺点：这种方法的不利之处就是其他人可以较易获得这个信息，然后对系统进行非授权访问。

（二）他持有的物品

这个物品具有独有的特征，可以区别其他物品（也就是所谓的惟一性），包括门卡、身份证、工作证等，以及可以在网络认证中使用的 SecurID、SIM 卡等。

虎符最早出现于春秋战国时期，当时采

用铜制的虎形作为中央发给地方官或驻军首领的调兵凭证，称为虎符。虎符的背面刻有铭文，分为两半，右半存于朝廷，左半发给统兵将帅或地方长官，并且从来都是专符专用，一地一符，绝不可能用一个兵符同时调动两个地方的军队，调兵遣将时需要两半勘合验真，才能生效。

虎符在古代战争中曾发挥了重要的作用，也发生了很多与其相关的故事。《史记》中记载，战国时期的公元前257年，秦国发兵围困赵国国都邯郸，赵平原君因夫人为魏信陵君之姊，乃求援于魏王及信陵君，魏王使老将晋鄙率10万军队救援赵国，但后来又畏惧秦国的强大，命令驻军观望。魏国公子信陵君无忌为了驰援邯郸，遂与魏王夫人如姬密谋，使如姬在魏王卧室内窃得虎符，并以此虎符夺取了晋鄙的军队，大破秦兵，救了赵国。

在《三国演义》第五十一回中，曹操因赤壁之战兵败北退，诸葛亮则趁南郡空虚，命勇将赵云夺城成功，并且俘获守将陈矫，取得虎符，然后以此虎符诈调荆州守军出救南郡，趁势又由张飞袭取了荆州。接着再用



同样的方法调出襄阳守军，乘机由关羽袭取了襄阳。诸葛亮仅凭一个小小的虎符，便将曹兵调开，兵不血刃就夺取了三处城池，而耗费许多钱粮、兵马的东吴周瑜却一无所获，如何不生气？由此也可见当时虎符作用之大。

优点：因物理接触人较少，安全性较高。

缺点：这种方法的缺点是容易丢失或被盗，从而导致未授权访问。

（三）他就是这个人

属于生物特征识别技术，包括指纹、掌纹、手形、视网膜、虹膜、DNA 等。这类凭据的安全性较高，一个因素是不易伪造，二是出现重合的机率非常小。例如，指纹重合的概念为百万分之一，辅以其其他证据，可以很轻易把目标人与其他人区别开来；DNA 的重合率为 1/1016，基本可以通过 DNA 鉴定来认定某人。

托米林拿出一包令人毛骨悚然的照片和资料，其中有一张从正面和左右侧面三个角度拍摄的头骨照片。托米林指着照片说：“这正

是林彪的头骨。”并解释说林彪的头部在战争中受过伤，其位置正好与头骨的伤痕相吻合，而且苏联保有林彪 1938—1941 年在莫斯科治病的详细病历，有关林彪牙科记录也与头骨的实际情况丝毫不差。

扎格沃兹丁补充说：“我们另一个鉴定方法，是用头骨对照了林彪生前的照片。克格勃的资料里有一张俯拍的免冠照片，清楚显示了林彪头部的伤痕。我们还把头骨照片和林彪过去的一些照片叠放，看到两者的轮廓完全重合。”

托米林还说，人们的耳廓如同指纹，一个人一个样，没有重复的，因而是鉴定身份的重要依据。当年从现场割下了那具女尸的一只耳朵，与叶群的有关资料对照，得出了相应的结论。

为了使鉴定头骨和耳廓的结论万无一失，克格勃的这个调查组，根据林彪病历中患过肺结核的记载，重返蒙古检验尸体。托米林记得那天正好是 11 月 7 日十月革命节，北风怒号，天寒地冻，他们挖出了林彪尸体，在其右肺确实发现钙化的硬块，与病历中的 X 光片一致。

优点：安全性很高。

缺点：操作麻烦，使用成本较高。

二、认证过程

根据凭据不同，认证过程也不尽相同，而在认证过程中，最主要的就是要确保凭据的惟一性或有限惟一性。

下面就来简单阐述与上面三种凭据对应的认证过程。

（一）他知道的内容

内容，哲学名词，指事物内在因素总和，与“形式”相对。世界上任何事物没有无形式的内容，也没有无内容的形式。内容决定形式，形式依赖内容，并随着内容的发展而改变。但形式又反作用于内容，影响内容，在一定条件下还可以对内容的发展起有力的促进作用。内容和形式是辩证的统一。

——百度百科

做个类比，内容与形式的关系相当于信息与数据，内容或信息等概念都是形而上的东西，我说不明白，但大家可以将它们理解为生物可以感知的东西。

举个例子，比如说军队在某地驻扎下来后，会通过口令来识别敌我，以防小股敌人侵入。在这个例子中，假设口令是“9628”，那么来自五湖四海的士兵在说“9628”时，口音肯定不一样，但只要内容是“9628”，而不是“你去死吧”之类的，就会被认为是正确的，是自己人。不过，前提是认证方是人而不是计算机之类的机器。极端一点，假如军中有外籍人士，他的回答是“nine six two eight”也是会被认可的；但如果认证方是计算机，它就没这么智能了。计算机

与人识别的东西在这里是不一样的，计算机只认识形式，而人可以认识形式抽象后的内容，显然更高级一些，这也就是所谓的“智能”吧。

说句题外话，很久以前，曾担心如果机器人统治地球后，我们怎么办呢？后来渐渐明白，机器人不是人，只是机器，它们没有爱恨情仇，没有金钱、权力欲望，没有生死离别，也就无所谓统治和奴役。如果出现机器人军队、机器人警察等暴力机器，那它的背后一定有一个高级生物体。

说了这么多只为强调计算机只能识别形式。具体到认证过程，一般涉及如下几个步骤：

1. 取原来存储的密码
 2. 与输入的密码做比较
 3. 根据比较的结果转向不同的处理过程
- 这三个步骤都有可能出现问题。

在第一个环节，最大的威胁是被拖库。

在第二个环节中，可能存在操纵比较的过程，从而绕过认证，最有名的例子就是SQL注入，这是认证过程中，约束条件最少，最有可能出现问题的地方，现在已有较成熟的防护方案。

在第三个环节中，可以操纵比较后的处理走向，举个例子就是初在 crack 时，一般会用到的“爆破”技术。不过这个环节中，可以实施的前提是可以修改认证服务器上的程序，而如果攻击者已经获取服务器上的修改程序的权限，那认证已经不是问题了。

“他知道的内容”认证过程中最大的问题是内容这个东西的惟一性很容易被破坏，即使我们可以确保在上述认证步骤中，服务器端没有什么问题，那密码仍可能被泄露，因为还存在客户端机器上的键盘记录器、网络上的 Sniffer、伪造的钓鱼页面等。而且密码泄露后还不容易被发现，不像你的大门钥匙丢了，即使捡到的人不做任何动作（或丢在某个角落里），你也会很快发现，而密码这东西，如果攻击者足够小心，他是能隐藏一段时间的。

（二）他持有的物品

物品一般是一个具有惟一特性的实物，它被认证机关赋予与被认证人相关联的属性。随着时代的发展，可用于认证的物品可分为三类：

一类是传统物品，比如说一代身份证、

早期的通行证等，它们一般是通过惟一的编码来确保惟一性。大家都学过“列宁与卫兵”的故事吧，这个故事本身就是此类认证过程的集中体现。

此类认证取决于认证人的经验与判断，与物品的特性有关，比如物品质地、色泽、大小、形状等。要想知道这类认证过程，可以看一下中央二套的“鉴宝”节目。

一类是通常存储电子信息的物品，比如说二代身份证、门卡等（有接触式与短距离非接触式），此类物品的认证过程为物品发送识别码，认证方将识别码与保存的数据进行比对，如果相符就会通过认证。

此类认证的最大威胁是 CLONE（复制）。

一类是具备信息交换的物品，比如我们用的 SecurID、手机+SIM 卡等。

此类物品的认证过程因认证实现方式的不同而不同。例如手机+SIM 卡的认证过程主要是通过 SMS 来传递，它的惟一性是由 SIM 卡来保证，不过，SIM 卡里的信息也存在被复制的可能性。除了通过 SMS 来传递认证信息外，现在也有一些厂商推出了与手机绑定的认证产品，比如手机密令、手机保令等。对于此类认证，较大的问题的手机丢失（被盗）后引发的冒用问题。

（三）他就是这个人

“他就是这个人”的认证过程主要是对与此人相关的生物属性进行认证。此类认证，一是认证过程较烦琐，二是对认证设备有要求，所以互联网应用中尚未见到用此类认证方式。但随着科技的发展，未来的某一天也可能会用到此类认证。

在此举几个例子，大家了解一下此类认证的大致过程即可。

李红平目前在福建开了一家服装店。联系上李红平后，这名从 9 岁起就失去母亲的女儿很是诧异。“我告诉她我们找到了她母亲，并且母亲还活得好好的，她当时根本不相信。”谭出洪说。后来谭出洪打电话回湖南给妻子，让她带着袁香梅给李红平打电话。

当李红平听到母亲叫她的名字时，童年的记忆便涌上心头，虽然对母亲的记忆早已淡薄，但听到母亲的声音时还是有一种感应。当听到母亲说她眼角有一块胎记的时候，李红平就深信不疑，电话那头的，确实是她失散多年的母亲。

《小小胎记让母女相认》

三、认证过程中的信任

信任问题是我们每个人、每一天都会面临的，不管我们有意还是无意，它一直充斥着生活的方方面面。

比如公司相信我们不是恐怖分子，所以不需要在公司入口架设安检设备；公司相信我们不会每天上班时间去购物网站、打网络游戏、看视频、听相声，所以不需要在公司网络出口架设网络监控设备；公司也相信我们每个人都睡得晚，起得更晚，所以会要求打卡。

但我们扪心自问，如果没有这些信任，一切将变得多么可怕，将造成何等的资源浪费。不过从社会学或心理学的角度来阐述信任问题远超出我的能力范围，因此，接下来，我还是把重点放在网络应用认证过程中的信任问题。

通过网络进行认证涉及几个问题，一是凭据要能通过网络传输，二是要保证凭据的有效性（惟一性、偶合概率），三是整个认证过程

中每个环节的信任问题。

凭据要能通过网络传输，是指凭据必须是数字化（电子化）的形态。那就表示实物不可用于网络认证。此外，还要考虑实施认证的便利性及成本问题。例如，从技术上讲，是可以将指纹用于网络认证的，但目前市面上尚未见有使用此类技术的公司。

保证凭据的有效性（惟一性，偶合概率），是指要尽量保证凭据是独一无二的。从理论上讲，这是无法做到的，但从实际出发，我们只要把重合的概率控制在一定的范围内即可。例如，我们前面提到指纹的重合率为百万分之一，DNA 的重合率为 $1/1016$ ，在这种情形下，我们基本可以认定它们的惟一性。我们来看一下密码的重合率，假设长度为 8 位，那么全数字（10 种）的密码重合率为百万分之一，26 个字母（大写或小写，26 种）的重合率为两百亿分之一，26 个字母混合（52 种）的重合率为 53 兆分之一，字母加数字（62 种）的重合率为 218 兆分之一，而字母 + 数字 + 符号（96 种）重合率为 $7.2 \times 1/10005$ 。所以一定的密码强度基本可以保证它的有效性。此外，还可以通过隐匿用户名、增加验

证码、连续输入错误锁定等方式来进一步保护凭据的有效性。

整个认证过程中每个环节的信任问题，是指确保从凭据的输入、传输及处理等环节保证凭据不被侵害。

输入环节的信任是指对客户端计算机运行环境的信任。因为利益的驱使，现在游戏盗号木马、网银木马十分猖獗，客户端运行环境基本不被信任，对安全性要求较高的公司会通过输入控件防范恶意软件盗取认证凭据，比如网银、第三方支付以及一些电商也在行动。

传输环节是指通过互联网传输凭据，因为互联网属于不可控因素，所以不被信任。所以为了防范凭据被盗取，一般采用加密的方式进行防护。

处理环节属于服务端的运行环境，是整个认证的核心所在，如果服务端不被信任，所谓的认证就失去了它的意义。所以我们要保证操作系统、处理程序等所有组件都是可信的。

有人可能质疑说，这些都是技术问题，并不是什么信任呀，是不是你故弄玄虚。确

实是这样，但是，即使在某些层面我们可以通过技术手段进行控制，但毕竟有些东西是我们不可控的，我们要么信任要么放弃。例如，微软的操作系统，曾流传过专为 FBI 留有后门，你能完全放弃吗？

列宁曾说过“信任是好的，但控制更好”，但在我们力有不足的时候，我们一定会有所信任。怎样均衡信任与控制，达到最优，其实已经不属于技术范畴了。

四、网络应用中常见的认证机制

各种凭据因其特性不同，因而有不同的安全等级。比如生物认证虽然也不乏通常胁迫、肢解等手段来破坏其有效性，但综合而言它的安全性最高，可实施代价太大，而且不方便，所以一般采用的地方较少，而且基本都是面对面的验证，没有通过网络验证一说（理论上可以通过网络认证，不过又增加了泄露的风险，丧失了生物认证应有的安全性），所以本文不讨论生物特征的认证技术。但不排除在未来的某一天，随着科技的发展，我们可能可以通过生物特征来方便地进行认证。

前面已经讲述除了凭据本身的属性影响

安全性外，对其进行验证（鉴别）的过程也会影响认证的有效性。接下来，我就来介绍几种常见的网络认证机制，看看认证过程是怎样鉴别凭据的。

（一）普通认证机制

普通认证机制在这里指简单的认证机制，因为找不到一个特技术的名字，只好以普通之名概括之。在网络环境下，普通认证机制主要包括三种。

明文认证机制

这个在 CSDN 被曝拖库前有不少商家使用，其用意如何只有当事人知道。它的认证过程是：

1. 客户端把密码明文发给服务器。
2. 服务器将密码与数据库里保存的密码明文进行比较。
3. 服务器通知认证成功或失败。

Hash 认证机制

这个以前比较常见，它的认证过程是：

1. 客户端把密码明文发给服务器。
2. 服务器将接收到的密码经过 Hash 后与数据库里保存的 Hash 串（在注册时将密码进行 Hash 处理后生成的字串）进行比较。

3. 服务器通知认证成功或失败。

Hash+Salt 认证机制

上一种认证机制有一个问题是，当数据库被拖库时，即使保存在数据库里的密码经过 Hash 处理，仍有很大可能被暴力猜测，在这种情形，开发者又开发了 Hash+Salt 认证机制，它的认证过程是：

1. 客户端把密码明文发给服务器。
2. 服务器根据用户名从“盐表”找到对应的“盐 (Salt)”（也可能是固定的“盐”），然后与用户的密码连接，并将连接后生成的字串进行 Hash 处理，然后将这个 Hash 串与数据库中保存的 Hash 串进行比较；
3. 服务器通知认证成功或失败。

这三种认证的方法从弱到强增加了保存在数据库中的密码的安全性，但对网络上传输的密码并无助益。面对此类风险，比较流行的做法是使用 HTTPS 协议。

（二）公钥认证机制

使用这种机制必须有一个第三方的 CA 为客户签发身份证明。客户和服务器各自从 CA 获取证明，并且信任该 CA。在会话和通讯时首先交换身份证明，其中包含了将各

自的公钥交给对方，然后才使用对方的公钥验证对方的数字签名、交换通讯的加密密钥等。在确定是否接受对方的身份证明时，还需检查有关服务，以确认该证明是否有效。

认证过程：

1. 客户端产生一段字串，然后对这段字串进行 Hash 处理。客户端再用用户的私钥对生成的字串进行加密，并将原始的字串和加密后的结果传送给服务器。

2. 服务器将收到的字串进行同样的 Hash 处理，同时也用发送方的公钥对加密的字串进行解密。如果解密后的字串和服务器自己产生的字串一致，那么接收者就可以相信对方的身份，因为只有发送方的私钥才能够产生加密后的字串。

3. 要向发送方验证接收方的身份，接收方根据自己的私钥创建一个新的数字签名，然后重复上述过程即可。

（三）挑战 / 应答的认证机制

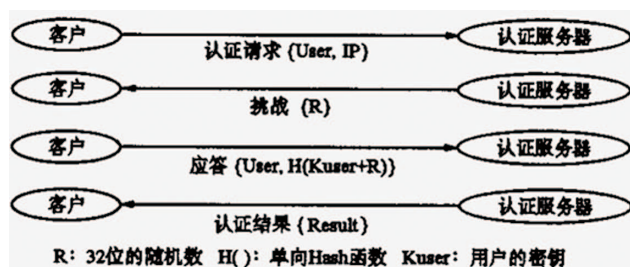
基于挑战 / 应答 (Challenge、Response) 方式的身份认证机制就是每次认证时认证服务器端都给客户端发送一个不同的“挑战”

字串，客户端程序收到这个“挑战”字串后，作出相应的“应答”。

认证过程：

1. 客户端向认证服务器发出请求，要求进行身份认证；
2. 认证服务器从用户数据库中查询用户是否是合法的用户，若不是，则不做进一步处理。
3. 认证服务器内部产生一个随机数，作为“挑战”，发送给客户端。
4. 客户端将用户名和随机数合并，使用单向 Hash 函数（例如 MD5 算法）生成一个字串作为应答。
5. 认证服务器将应答串与自己的计算结果比较，若二者相同，则通过一次认证，否则认证失败。
6. 认证服务器通知客户端认证成功或失败。

以后的认证由客户端不定时地发起，过程中没有了客户端认证请求一步。两次认证的时间间隔不能太短，否则就给网络、客户端和认证服务器带来太大的开销；也不能太长，否则不能保证账号不被冒用。



五、该如何选择

作为用户，我们并不能决定商家采用何种认证机制，但我们可

以用脚投票。

比如说，如果你想开通一家银行的网上银行，它只用密码认证，并且可以无限额转账，那么你敢用吗？我是不敢。

但作为商家，并不是安全性越高越好，还要综合考虑经济性与便利性，最主要的是要根据业务的需要。

比如一个普通的论坛，除了登录密码外，没必要每次发帖时再要求输入二次密码。如果是采用货到付款，那么商家只用登录密码进行认证也是可以接受的，因为即使密码被盗，也不会损失什么；而如果账号上有余额或礼品卡的话，那么只用登录密码进认证就显得有些单薄了。

下表是我个人从安全性、经济性、便利性对各类认证方式的评分（1为最高分），仅供参考。

认证方式	安全性	经济性	便利性
登录密码	0.3	1	1
二次密码（支付密码）	0.4	1	1
数字证书	0.8	1	0.5
手机 （手机短信、手机密令）	0.7	0.7	0.7
密保，SecurID	0.7	0.3	0.5
USB Key （数字证书）	0.9	0.3	0.5

参考文献

1. 《在大漠那边：林彪坠机真相》
2. 虎符，百度百科。http://baike.baidu.com/view/15995.htm

手机银行的几个典型风险

北京分公司 蔡昆

关键词：手机银行 安全风险 信息安全

摘要：手机银行正越来越被大家所熟悉，本文描述了在手机银行的安全评估工作中总结的几个典型风险，对风险进行了分析和探讨，提出“合规为前提、业务安全为关键、技术安全为基础、管理为保障”的安全防护思路 and 手机银行安全防护结构图。

引言

手机银行以其使用便携、手续费低廉，而越来越受到人们的喜爱，但仍有相当部分的人因对手机银行安全性的担忧而不敢使用。笔者参与过多家银行的手机银行系统的安全评估，总结归纳了几个较为典型的风险，希望能够对降低手机银行的风险，增强人们的信心有所帮助。

一、手机银行的限制和分类

(一) 手机银行的限制

手机银行是运行在手机终端上的应用，

与运行于电脑上的网上银行相比，有如图 1 所示的一些限制，从而影响了一些传统安全方式的使用：



图 1 手机银行的限制因素

- 手机的计算能力低，限制了对非对称加密算法的使用。
- 手机的屏幕远远小于电脑，无法在操作的同时提供必要的安全提示。
- 手机的硬件缺乏，没有 USB 口，限制了对 U 盾和数字证书的广泛使用。
- 手机浏览器比电脑的浏览器功能少，不支持控件（如密码控件）。
- 手机的网络速度大大低于电脑的上网速度，特别是 GPRS/CDMA 等 2G/2.5G 通信，限制了应用协议的使用。

(二) 手机银行的分类

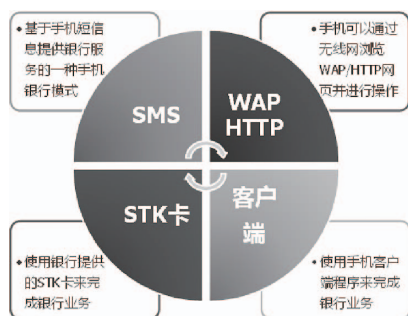


图2 手机银行的分类

手机银行由于其独特性而有多种实现方式，不同方式的手机银行有不同的安全风险，所以在描述手机银行的风险前先简单介绍下手机银行的分类。目前，根据实现方式主要有以下几类，如图2所示：

- 短信银行

通过手机短信来实现银行业务，客户和银行通过手机短信交互信息，安全性低，所以主要做查询业务。

- STK 手机银行

又称手机银行卡，靠 STK 卡（手机银行卡）提供的加密短信来实现银行业务，安全性高，其前提是需要客户将 STK 卡粘在手机 SIM 卡上，银行服务菜单写在 STK 卡中。

- WAP/HTTP 手机银行

跟网上银行类似，通过网页提供银行业务，安全性高。其中 WAP 适用于上网速度低的情况，HTTP 适用于 3G 高速网络。

- 客户端手机银行

安装银行提供的客户端软件，通过客户端软件访问银行实现手机银行的功能，安全性高。根据不同的手机平台，分为 iPhone、Android、Windows Mobile、KJAVA、MTK 几种，其中以 iPhone 和 Android 为最多。

- 其他

其他还有 USSD、BREW 等实现方式，但现在已经很少使用，所以本文不再讨论。

浏览器的客户端程序，网上银行一般是通过浏览器访问的，而手机银行使用通过短信、浏览器、客户端程序等多种方式，客户端程序又根据不同的平台而分为多种版本，手机银行的客户端程序也是比较受关注的风险集中地。

手机银行的网络通信涵盖了有线和无线通信，包括了 GSM、3G、TCP/IP、WiFi 等，使用了短信、加密短信、TCP、WAP、HTTPS 等通信方面，风险面比银行的标准 HTTPS 通信要大。比如短信手机银行的所有交易信息（包括金额、密码等信息）会在移动运营商处留有记录。

由于手机银行应用相对较新，所以在后台内部管理、业务等方面还不如网上银行的完善，有一些诸如角色设置、权限设置、日志审计等方面的问题。手机银行服务器端还使用了加密机、密钥卡、读卡器等设备，也引入了一些风险。

本文将对 10 个典型的安全风险进行描述，这 10 个风险是：

- 手机银行卡合规风险
- 非法下挂账号风险

二、手机银行安全风险与防护

（一）手机银行的安全风险

由于手机银行比网上银行的种类多，实现方式多，所以比网上银行有着更多更复杂、涉及面更广的风险，无论是在客户端、网络通信，还是服务器端。

在客户端方面，除了手机本身的风险，还包括手机卡、手机短信、手机客户端程序、手机浏览器的风险。除了招商银行使用内嵌

- 通信风险
- 内部管理平台滥权风险
- 短信延时丢失风险
- 客户端接入风险
- 密码猜测风险
- 服务器私钥泄露风险
- 客户通信密钥泄露风险
- 锁定无通知风险

(二) 手机银行的安全防护

基于上述的风险分析，我们提出“合规为前提、业务安全为关键、技术安全为基础、管理为保障”的安全防护思路和手机银行安全防护结构图，如图 3 所示。

图 3 手机银行安全防护结构图

业务	申请 临柜业务 内管	限额 转账支付 教育
技术	客户端 手机环境 短信	手机卡 客户端程序 安全辅助工具
	网络通信 服务器端	GSM 3G TCP/IP WIFI WAP HTTP HTTPS 蓝牙 运营商节点 物理层 网络层 系统层 应用层 数据层
管理	机构 制度 人员 建设 运维	
合规		

因篇幅所限，本文不再对安全防护结构

图进行详细阐述，下面让我们来看一下典型的手机银行风险。

三、手机银行卡合规风险

(一) 适用手机银行类型

STK 手机银行

(二) 风险描述

STK 手机银行，需要在客户的手机安装一个小卡片，此卡片被称做手机银行卡，手机银行卡粘在客户的手机 SIM 卡上，与手机 SIM 卡一起插入客户的手机卡槽中。笔者在对安全评估过程中发现，手机银行卡并未取得国家规定的强制的资质证书，这对手机银行业务的合规提出了挑战。

在中华人民共和国国家质量监督检验检疫总局、中华人民共和国国家认证认可监督管理委员会、中华人民共和国财政部联合发布的《关于部分信息安全产品实施强制性认证的公告》和《关于调整信息安全产品强制性认证实施要求的公告》中，要求“凡列入本强制性认证目录内的信息安全产品，未获得强制性产品认证证书和未加施中国强制性认证标志的，不得出厂、销售、进口或在其他经营活动中使用。”手机银行卡是标准的 CPU 智能卡，遵循 ISO7816-4 标准，属于第一批信息安全产品强制性认证目录中的“5) 智能卡 COS”，所以手机银行卡应取得信息安全产品认证证书。

他经营活动中使用。”手机银行卡是标准的 CPU 智能卡，遵循 ISO7816-4 标准，属于第一批信息安全产品强制性认证目录中的“5) 智能卡 COS”，所以手机银行卡应取得信息安全产品认证证书。

(三) 风险分析

手机银行卡应取得国家信息安全产品强制认证证书，不取得此证书，存在很大的合规风险，监管部门可能会因为手机银行卡的不合规，而对手机银行业务进行处罚，对银行的声誉、客户的信任直接带来负面的影响。

(四) 风险应对

采购合规的手机银行卡，或敦促现有手机银行卡厂商取得相关信息安全产品认证证书，消除此合规风险。

四、非法下挂账号风险

(一) 适用手机银行类型

短信银行、STK 手机银行、WAP/HTTP 手机银行、客户端手机银行

(二) 风险描述

手机银行都允许客户多个账号开通手机

银行，下挂账号功能正是应此需求而设置的，让客户可以自己下挂其他账号。但当客户的手机银行被盗，而下挂账号功能的审核又不严，就可能导致账号被非法下挂，从而导致客户进一步的资金损失。

(三) 风险分析

不少人在同一银行有多个账号，为安全起见，只会将其中一个账号开通手机银行（通常是资金相对较少的账号），以免因手机银行被盗而损失其他账号的资金。所以对于下挂账号的功能必须严加限制和审核，当客户的手机银行被盗或被入侵，不至于被入侵者非法下载客户的其他账号，而造成其他账号的资金也遭遇损失。

(四) 风险应对

对下挂账号的功能进行限制和严格审核，如客户自助下挂账号最多只能进行查询而不能进行任何转账支付功能，对于下挂的账号需要进行转账支付时，必须到柜台办理。

除了下挂账号，手机银行还有一些高风险业务是需要客户到柜台进行办理的，如：

- 变更手机号码
- 普通用户（自助注册手机银行的用户）申请同名下的账户转账、协议账户转账功能
- 安全辅助工具（如动态口令）的申领、更换、解除挂失
- 重置手机银行登录密码、交易密码

五、通信风险

(一) 适用手机银行类型

WAP/HTTP 手机银行、客户端手机银行

(二) 风险描述

手机银行直接涉及客户敏感信息、客户资金，所以对于通信的加密和保护是必要的，否则就可能被窃听和篡改，导致客户信息的泄露和资金的损失。

当前最可靠的加密方式是 HTTPS 加密，而在实际的手机银行实现中，出于对性能等方面的考虑，有一些手机银行根本没有加密，有一些手机银行没有采用 HTTPS 加密方式，而是采用自定义的加密协议，这样通信的安全程度如何就看自定义加密协议是否可靠了。

(三) 风险分析

手机银行的使用涉及无线局域网、移动无线网、Internet 互联网等多种网络环境，也面临这些网络环境的复杂威胁，所以通信的加密和加密的可靠性至关重要。在笔者评估过的手机银行中，发现过以下几种通信问题：

- 通信未采用加密协议，数据未加密。
- 自定义加密协议易受重放攻击，未能采用时间戳、序列号、多次握手等技术防范重放攻击。
- 未采用消息认证码（MAC）来保护消息的完整性。

(四) 风险应对

采用 HTTPS 加密协议是目前主流可靠的风险应对方式，虽然对手机本身有了一定的要求，但在现在手机硬件性能快速提高的背景下，将不再成为阻碍的因素。

六、内部管理平台滥权风险

(一) 适用手机银行类型

短信银行、STK 手机银行、WAP/HTTP 手机银行、客户端手机银行

(二) 风险描述

手机银行的内部管理平台负责客户管理、商户管理、操作员管理、统计分析等，通过内部管理平台能够直接对客户进行操作，所以内部管理平台的权限控制至关重要。如果权限控制不严，内部人员可能利用权限管理的漏洞并造成客户的资金损失。

(三) 风险分析

笔者在安全评估过程中，遇到过两种权限管理的漏洞，一种是角色设置不合理，一种是没有复核机制。

角色设置不合理是指没有按照三权分立的原则设置管理员角色、操作员角色、审计员角色，普遍缺乏审计员角色，更有甚者，管理员身兼管理职能和业务操作职能，职责划分不明确。比较典型的是系统设置了两个最高权限级别的管理员，如“总行系统管理员”，这两个总行系统管理员拥有内管平台所

有权限，包括管理职能的权限和部分业务职能的权限，而实际上总行的管理员不应具有业务职能的权限，顶多只有统计查询权限，这就存在很大的滥用权限的风险。

另一种漏洞是没有复核机制。复核机制是降低操作风险的一种有效机制，在银行业中应用普遍，网上银行的内管平台都设置了此机制，而由于手机银行业务较新，此机制还不在网上银行业务被普遍采用。如果没有复核机制，具有较高权限的管理员，如总行系统管理员，可能在没有监督情况下进行违规操作，如创建操作员，并以操作员身份对客户进行操作，进而导致客户的资金损失。

(四) 风险应对

根据三权分立原则设立管理角色、操作角色和审计角色，根据最小授权原则，控制角色的权限范围，防止进行越权操作。设置复核机制，以实现不同管理员之间的监督和制衡。

七、短信延时丢失风险

(一) 适用手机银行类型

短信银行、STK 手机银行、WAP/HTTP

手机银行、客户端手机银行

(二) 风险描述

手机银行中，短信的应用已经相当重要，甚至已经达到了不可或缺的地步，短信银行和 STK 手机银行这两种手机银行都以短信为通信方式的，WAP/HTTP 手机银行和客户端手机银行都以短信动态口令作为转账支付的主要确认手段，而且短信动态口令都有严格的时效限制。在这种情况下，短信的延时或丢失，会直接造成交易的失败，影响客户的使用信心。笔者在安全评估时，曾有银行反应短信的延时或阻塞率有时能达到 10%。

(三) 风险分析

对于银行来说，短信延时或丢失的原因主要有两个方面，一是短信网关问题，一是移动运营商问题。

银行采用短信网关下发短信，而短信网关存在短信阻塞、宕机等常见问题，特别是宕机问题，会造成短信的延时和丢失。

移动运营商的短信处理能力是有一定的限制的，特别是在节假日等高峰时间，短信的阻塞现象很突出。短信在移动运营商的保

存时限为 48 小时，超过时限短信就失效，不再发送，所以与移动运营商对短信送达的协议约定很重要。

(四) 风险应对

采用技术手段保障短信网关的稳定运行，如提高硬件处理能力、采用短信失败重发机制等。与移动运营商签定协议，对短信的送达时效进行约定，保障短信的送达。

八、客户端接入风险

(一) 适用手机银行类型

WAP/HTTP 手机银行

(二) 风险描述

手机银行系统未对手机终端接入进行严格限制，客户除使用手机终端上的浏览器通过移动网络访问手机银行之外，还可以通过电脑上的浏览器（如 IE、Firefox 等），通过互联网直接访问手机银行。

(三) 风险分析

手机银行系统考虑到手机终端的相对封闭性，对客户端提供了弱于网上银行系统的安全控制，例如图形验证码更为简单、无客户端密码安全控件等。相对弱化的客户端安全控制措施，对于手机终端来说，尚可接受，但是如果对于电脑 PC 接入来说，其信息泄露风险要明显高于网上银行系统，允许电脑 PC 接入访问手机银行系统，相当于提供了安全级别降级的网上银行系统登录。

(四) 风险应对

在客户登录过程中，通过技术手段增加对于手机终端的认证，

仅允许手机终端接入访问手机银行系统，屏蔽直接互联网访问可能造成的风险。

九、密码猜测风险

(一) 适用手机银行类型

STK 手机银行、WAP/HTTP 手机银行、客户端手机银行

(二) 风险描述

密码是手机银行登录和转账支付的主要凭证，其重要性不言而喻，密码被猜测，直接可能导致敏感信息的泄露和资金的损失。现在手机银行系统在登录时，都对密码的输入错误次数进行了限制，密码输入错误连续达到一定次数会临时锁定登录，但很多手机银行系统对其他需要输入密码的功能（如转账支付）却并没有进行密码错误次数的限制，带来较大的密码猜测风险。

(三) 风险分析

手机银行的密码分为登录密码、支付密码、卡密码几种，其中卡密码的校验和锁定一般由银行的核心系统实现，登录密码和支付密码的校验是由手机银行系统实现。手机银行系统中需要对密码校验的位置包括登录、转账支付、修改密码等处，对登录时密码错误次数的限制都能做到，但对其他密码校验功能往往缺乏相应控制。大家知道，手机在输入密码时会将密码字符显示出来，手机银行的密码输入也同样会显示出来，而手机经常在公众场合使用，所以手机银行登录密码的泄露可能性是很高的。倘若有人偷看到了登录密码，然后在转账支付页面猜测支付密码，一旦猜测成功，就不仅仅

是敏感信息的泄露，而是客户直接的资金损失。

(四) 风险应对

梳理手机银行系统中所有存在密码校验的功能，对密码的错误次数进行限制，如密码连续错误五次进行当天的锁定、密码连续错误十五次永久锁定等。

十、服务器私钥泄露风险

(一) 适用手机银行类型

STK 手机银行、WAP/HTTP 手机银行、客户端手机银行

(二) 风险描述

手机银行客户端与服务器端的通信加密，都要采用最安全的非对称加密算法，通常的会话建立流程是：客户端生成只在当前会话有效的会话密钥种子或会话密钥，然后将密钥种子或密钥用服务器的公钥进行加密传给服务器，服务器收到后用自己的私钥解密，随后双方使用会话密钥进行加密通信（对称加密）。

可以看出，这个算法是无懈可击的，但这是基于私钥是绝对安全的假设。私钥是非对称加密算法的基础，私钥一旦泄露，算法就是个摆设，跟明文传输没多大差别。

(三) 风险分析

手机银行的服务器私钥有以下几种主要的保存方式：

- 以文件形式保存在安全网关（专用硬件）上
- 以文件形式保存在服务器上
- 硬编码在服务器上的程序中

以上三种方式都存在，都比较常见。从安全性角度考量，第一种因为专用硬件本身相对封闭漏洞较少所以最安全，后面两种都容易被内部人员轻易获取，或者因为被外部入侵而泄露，或者被开发和实施人员泄露。其实最安全的私钥存放方式是保存在加密机中，加密机中保存的密钥是不可导出的，所以 PKI 公钥基础设施的 CA 私钥都是保存于加密机中。遗憾的是，笔者至今没有见到哪家银行的手机银行服务器私钥存放在加密机中。

(四) 风险应对

将服务器私钥存放在硬件（安全网关、加密机）中是当前最安全的保存方式，不仅可以防范外部人员的非法获取，也可以防范内部人员。

十一、客户通信密钥泄露风险

(一) 适用手机银行类型

STK 手机银行

(二) 风险描述

通信密钥的泄露会导致客户敏感信息的泄露。STK 手机银行的通信密钥主要有两种泄露途径：

- 通信密钥被内部人员窃取
- 通信密钥被破解

(三) 风险分析

- 通信密钥被内部人员窃取

STK 手机银行的客户通信密钥最初是由加密机产生的，然后导

出到一个 IC 卡大小的卡片（一次保存上百个密钥）中，再从这个卡片中写进手机银行卡中。所以保存上百个密钥的卡片的保管和使用就至关重要，存在很大的卡片流出、密钥读出的风险。

有的手机银行系统为了消除这种风险，就采用了初次使用即更换的方法，很有效。就是说，初始的客户通信密钥只在第一次使用（即注册时）有效，使用后，由手机银行服务器通过加密机新生成一个通信密钥，并将新的通信密钥自动写进客户的手机银行卡中，原来初始的通信密钥被作废，即便泄露也没有什么影响了。

■ 通信密钥被破解

虽然有的 STK 手机银行（又称手机银行卡系统）宣称采用非对称加密算法对通信进行加密，但实际上现在的 STK 手机银行的通信几乎都只是采用对称加密算法，这主要是出于对手机性能方面的考虑。STK 手机银行的通信密钥是每个客户一个通信密钥，通信密钥基本都是在制卡（贴片）时就已经写在手机银行卡中了。大家都知道，对称加密算法可以通过监听加密数据包进行分析，是可以破解出密钥的，破解的效率主要取决于密钥的长度和足够的数量。而在实际中，客户的通信密钥一般不会更换，这就为密钥的破解留了充足的时间。所以通信密钥的定期更换就变得很重要。

(四) 风险应对

通信密钥采用初次使用即更换的方法，防范内部人员窃取密钥。

通信密钥定期更换，防范密钥破解。

十二、锁定无通知风险

(一) 适用手机银行类型

WAP/HTTP 手机银行、客户端手机银行

(二) 风险描述

手机银行出于安全考虑，都设置有账户锁定机制，但账户锁定后却没有通知手段来通知客户，给客户造成很多困扰。

(三) 风险分析

手机银行的登录，都有密码错误的锁定机制，比如：密码连续错误 5 次就当天锁定此账号，到第二天自动解锁；密码连续错误 15 次就永久锁定此账号，只能到柜台进行解锁。若恶意人员尝试猜解口令，很可能导致账号被锁定，而真正的客户却毫无所知，等到使用时才发现已经被锁定，影响客户的正常交易，从而影响银行的声誉。

(四) 风险应对

设置锁定通知机制，在客户账号被锁定后通过短信进行通知。

十三、总结

最后笔者还想提一下手机银行的载体——手机本身的风险。现在人们对电脑的各种病毒、木马等风险印象深刻，但对手机本身的风险认识还不够。有人说“手机就是移动的电脑”，PAD 就是明证，手机也会慢慢向电脑的智能方向发展，手机的风险甚至多于电脑的风险，这就需要引起我们的关注。

手机银行相对来讲是新兴业务，正处于不断发展和完善过程中，还有很多安全风险需要我们去关注，笔者在此抛砖引玉，与大家共同探讨。

银行信息安全管理探讨（二）

行业技术部 徐一丁

关键词：银行 信息安全管理

摘要：接上期（一），本期主要从安全人员视角来讨论几个常见的安全管理问题。

安全人员与其他 IT 团队应如何分工？

名为安全人员，在大多数银行中，安全人员并不是要直接解决所有安全问题，很多技术问题是网络、系统、应用等人员直接解决的，如打补丁、实施网络访问控制策略。



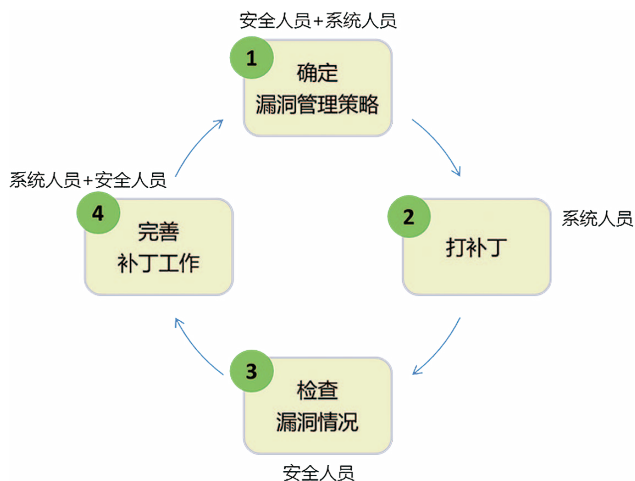
安全人员并不能掌握所有知识，通常需与 IT 相关人员一起讨论其相关部分的安全策略，由 IT 相关人员实施，安全人员评估与复核，如下面的系统漏洞管理流程中：

1. 系统漏洞的管理策略由安全人员牵头来组织，安全人员提出漏洞管理的原则和要求，如“为了安全起见，所有的重要系统都应打补丁”，系统管理人员会一起讨论，在哪些系统上应打什么补丁，哪些系统不适合打补丁，可以采取什么方式加强等。
2. 系统人员根据讨论好的方案打补丁。
3. 安全人员利用漏洞扫描器，结合人工

手段检查系统的漏洞是否都按照步骤 1 中讨论出的方案得到了执行，发现那些不符合项，如某个补丁没有打。也有可能发现步骤 1 中补丁方案的一些不足之处。

4. 系统管理人员根据检查结果，改正步骤 3 中发现问题，如某个补丁没有正确安装。双方还会一起讨论补丁管理策略和方案的不足并加以改善。

以上的工作应形成闭环，不断适应系统环境和各方面情况的变化。网络、系统、应用开发等其他各方面的安全工作也类似，都是安全人员与相关人员一起配合来完成的。



建议以增强安全人员实力为主，领导层面的政策与支持为辅。

安全人员应知道，其他团队有自己的专职工作与相应绩效。网络人员负责网络正常运行，系统人员负责主机与数据库的正常运行，从他们的角度看，安全工作不一定是必须的，甚至可能是在添麻烦，“以前不做安全工作也没出事”、“上了 Web 应用设备网络变慢了”……其他团队有配合与支持安全管理工作、共同实现安全目标的义务，但他们还是最关心自己的绩效，这是人之常情。

在本机构的 IT 系统正常运行和实现安全目标之间找到最佳的平衡点，是安全人员的关键职责。安全人员应充分理解其他团队的立场与想法，与大家共同讨论安全措施与方案，尽量少地影响原有 IT 运行环境；大家都应确认和接受不得不带来的影响，并在今后寻找消除负面影响的办法。

安全人员应注意加强自己的积累，讨论和沟通的前提是双方的知识经验对等。在很多银行安全人员说话没人听，通常是由于安全人员水平低，提不出合理的、可落地的建议，发现一个问题组织讨论时，很轻易地就被系统或网络人员挑出毛病，驳回安全人员的想法。这样的事情多了，安全人员做事情的信心建立不起来，在 IT 部门就没什么话语权，做事就更缩手缩脚，就更获得不了经验……这是个恶性循环。

安全工作行之有效的银行，无一不是安全团队或人员经验丰富，在 IT 部门有较高认可度的，而实力是获得大家认可的有效保证。

安全管理工作也要争得领导层的授权与支持。有了领导决策，加上合理的实施方案，安全工作就大有可为。

安全设备应由谁来管理?

安全设备在每个不同的机构，都应根据实际情况来确定日常管理者，以合理地平衡业务运行需要和实现安全管理为目标。

与系统相关层面的运行操作结合较紧密的安全设备，推荐由相关工作的负责人员来管理，如防火墙已经是构建网络架构的基础设施，管理与配置涉及更多网络专业知识，与网络运行环境的结合也很紧密，适合由网络人员管理。

Web 应用安全和上网行为安全类设备，安全人员或运行监控人员负责都很正常。表 1 给出了一些参考建议。

安全管理工作，其他团队或部门不配合怎么办?

表 1 安全设备与管理人员对照表

设备名称	安全人员	运行监控人员	系统/数据库人员	网络人员	应用开发人员/维护人员
流量分析与清洗设备				√	
防火墙				√	
Web应用防火墙	√	√			
Web应用扫描器	√				
安全审计设备	√				
堡垒机	√				
认证与授权系统			√		
网站内容安全设备	√	√			
上网行为安全设备	√	√			
VPN接入服务器				√	
网络加密机				√	
入侵检测系统		√			
入侵保护系统		√			
补丁服务器			√		
漏洞扫描器	√				
安全配置核查设备	√				
应用加密机					√
应用认证与授权系统					√
签名/验签服务器					√
动态口令服务器					√

而取得领导的支持仍然要靠安全人员自己。随着监管工作的日益推进和银行安全风险意识的加强，领导层也知道开展安全工作的必要性了。但给安全人员什么权限，给到什么程度，要由每个银行自己确定，这又很大程度上决定于安全人员的水平。没有人愿意在沙滩上建立城堡，领导也不会把重要的工作托付给一个不可靠的员工，如果由于安全人员的

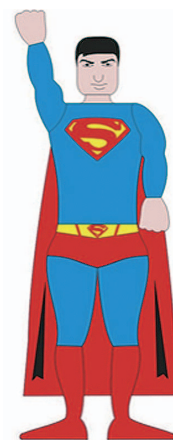
水平不足导致工作出现问题，那么本来正确的授权与决策也可能变成一个错误。

所以安全人员自身的综合实力决定了安全工作的效果。加强自己的专业水平，勤练内功，换位思考理解其他团队的想法，找出最适合本机构的安全方案，给团队和领导以信心，这是安全人员的必修课。

安全人员应具备哪些知识、能力？

安全问题往往牵涉很多方面的工作，安全人员应是个“通才”，从知识上看：

- 掌握至少一门安全管理最佳实践，如 ISO27001。
- 了解各类安全技术和事项。通常包括



风险评估方法、物理与环境安全、通信与网络安全、主机与系统安全、应用安全、认证授权、访问控制、加密、业务连续性和灾难恢复、安全审计等。

- 在网络、主机、应用开发等某一方面有深入的实践与理解，精通某一方面的安全技术。

- 对内部 IT 环境中的各个方面都有一定的理解与认识，从物理环境、网络、主机、数据库、Web 应用、业务应用都应熟悉。

- 作为银行人员，应对银行风险管理与控制理论非常熟悉。

安全人员不可能掌握所有的 IT 知识，但一定要能和所有的 IT 人（甚至非 IT 人员）沟通交流，以上的知识是安全人员与其他人员沟通的基础。

安全人员可以向诸葛亮学习，首先“观其大略”，把这些知识与情况通盘了解后，深入思考这些知识内存的逻辑联系，建立自己的信息安全架构和方法论，再到自己负责的工作中去实践、验证、完善。整体架构和全局视角非常重要，这是其他方面人员通常无法达到的层面，也是大部分的银行安全人

员职业发展的“独门绝技”。

同时，应当在网络、主机、应用开发等与 IT 业务运行密切相关的某一个领域有深入的实践与理解。如果说全面的知识理解是基础，那么这个领域的深入是安全人员登高远望的高楼。在一个领域高屋建瓴、融会贯通，能够使安全人员一通百通，因为他已经通过这个修炼过程掌握了基本的方法论。掌握了基本方法论，其他方面的事情即使不亲自动手做也能看得很清楚，能够有效把控安全的格局与方向。

银行安全人员也可以考虑参加一些有价值的认证，如：

- CISSP : Certified Information System Security Professional, 国际注册信息系统安全专家，由国际信息系统安全认证协会 ((ISC)2) 组织和管理，是目前全球范围内最权威、最专业、最系统的信息安全认证。

- CISP : 中国国内的信息安全专家认证，英文为 Certified Information Security Professional (简称 CISP)，系经中国信息安全产品测评认证中心实施国家认证，系国

家对信息安全人员资质的最高认可。知识领域与国际认证的 CISSP 很相似。

- ISO27001LA : ISO27001 主任审核员。适合信息系统管理人员、企业决策管理层、参与信息系统审计、管理体系规划和认证的人员，其他对信息安全管理及 ISO27001 知识感兴趣的人员。

- CISA : (Certified Information Systems Auditor 简称，中文为国际信息系统审计师) 认证是由信息系统审计与控制协会 ISACA(Information Systems Audit and Control Association) 发起的，是信息系统审计、控制与安全等专业领域中取得成绩的象征。CISA 认证适用于企业信息系统管理人员、IT 管理人员、IT 审计人员或信息化咨询顾问、信息安全厂商或服务提供商、其他对信息系统审计感兴趣的人员。特别适合银行内部的 IT 审计人员。

以上推荐的认证都是国际国内公认的权威认证，银行安全人员可以通过系统性地学习掌握这些知识，获得资历的积累，同时也可以学以致用，把学到的知识在银行安全管理中相互印证，达到更高的水平。

几个常见的DDoS botnet及其特点

核心技术部 刘亚 周大

关键字：botnet C&C 通信 DDoS 攻击 样本 通信协议

摘要：利用 botnet 发起 DDoS 攻击依然常见，本文从 bot 种类、C&C 通信协议、DDoS 攻击类型和服务器活跃情况等角度，对我们在过去一年里跟踪过的几十个大陆地区的 DDoS 类型的 botnet 进行了总结。

引言

以 botnet 为平台发起各种形式的攻击，比如 DDoS 攻击、垃圾邮件发送、信息窃取等，这种现象已存在多年，并且依然受黑产界的青睐，如果关注这方面的新闻会发现几乎每天都有相关报道出现。因为语言、文化的差异以及网络管制等原因，不同 botnet 之间往往具有比较大的差异，体现在规模、地理分布、“盈利模式”和技术手段等方面。比如在欧美地区比较活跃的 ZeroAccess、Zbot、Kelihos 等 botnet 在中国大陆地区就比较少见 [1][2]；再比如欧美的 botmaster 常将垃圾邮件发送作为生财手段，但很少见到中国大陆地区的 botnet 采取类似的盈利模式，倒是出现了不少游戏盗

号、DDoS 攻击类型的木马。

如果从技术上分析，会发现差别更大：一些“高端”的 botnet 已经开始采用诸如 FastFlux、P2P、公钥认证等这种以往只在正规的大软件系统中才会采用的复杂技术，以提高其健壮性和隐蔽性，而且会在客户端方面采取各种奇技淫巧（比如 bootkit 等）来提高保护等级 [3]，但也有一些 botnet 的客户端简陋到只是集成了有限的几条远控指令，连加壳保护都没有。

观察已有的 botnet 并寻找它们之间的共性是建立分类依据的重要手段，可以帮助我们更好地检测和防御通过 botnet 发起的攻击。本文将以实际数据为基础，对近半年来我们曾检测到的在大陆地区活跃的 13 种、99 个 botnet 做一些总结，分享一下我们的一些发现。

这些 botnet 均为 DDoS 攻击类型。

一、bot 客户端总结

bot 指 botnet 的客户端, 在被控主机(俗称僵尸机器、肉鸡)上运行, 负责接收和执行 botmaster 的指令, 通常被称为病毒、木马 (Trojan) 或者恶意软件 (malware), 与其对应的是由 botmaster 运行的控制服务器, 负责分发指令, 常被简称为 C&C 服务器。对 botnet 的分类通常是依据 bot 客户端, 比如基于 bot 家族来源或者所使用的 C&C 协议来划分, 我们只依据 C&C 协议对 bot 进行分类, 如果两个 bot 的 C&C 协议相同, 即使它们的 C&C 服务器 IP 或者端口号不同, 或者家族来源不完全一样, 也会被认为是同一种 bot。下面介绍我们对所检测到的 botnet 客户端的一些观察和总结。

1.1 bot 种类相对集中

前面提到, 我们共检测到过 99 个活跃的 DDoS botnet, 实际的 bot 样本近 1000 个, 但依据 C&C 协议进行分类后发现只有 13 种 bot, 也就是说这 99 个 botnet 的控制者先后为他们的 botnet 分发了近 1000 个

Md5 各不相同的 bot 样本, 而这些样本实际上都只是 13 个 bot 的变种而已, 有的可能连变种都算不上, 只是在每次传播时修改了样本中几个无关紧要的字节使 MD5 值发生变化而已。这说明实际中的 bot 种类远少于 bot 样本的数量, 新类型的 bot 没有那么多。

表 1 bot 家族分布

Bot 名称	出现次数
darkshell.c.2	44
darkshell.c	6
Nitol	14
artemis/storm.b	13
storm	6
neglemir	5
flyboy	2
Netbot	2
nxs	2
storm.a	2
jjjsfgv	1
Jrq	1
oxoddos	1

表 1 是我们对这 13 种 bot 的使用统计,

发现被使用最多的是一种名为 darkshell.c.2 的 bot (darkshell.c 是其早期版本), 有 50 个这样的 botnet, 占有 botnet 总数的一半以上, 这引起了我们的注意。调查后发现原来 darkshell.c 这种 bot 不但具有相对全面的 DDoS 攻击功能, 更重要的是其作者开放了源代码, 任何人都可以免费获取, 这无疑是其出现频率最高的主要原因。与此类似的是后面将要介绍的 RAT 软件 gh0st, 也是因为其源代码开放导致使用频率非常高, 出现了各种变种。再联想到来自俄国的臭名昭著的 bot 软件 zeus, 也是因为源代码被泄露以至于很快就出现了各种变种, 有人甚至为其添加了 P2P 和 DGA 功能 [6]。由此似乎可以得出结论, 开放源代码的 bot 更受 botmaster 的欢迎。

1.2 C&C 通信基于 TCP

对这 13 种 bot 进行分析发现它们的 C&C 协议都基于 TCP, 而且普遍使用长连接, 通过 TCP 的 KeepAlive 机制实现存活性检测, 未发现基于 UDP 的 C&C 协议。猜测这么做的原因可能是为了简化编程, 因

为如果要想实现基于 UDP 的 C&C 协议，需要自己处理丢包、乱序等各种繁琐问题，远比使用 TCP 来的复杂。

1.3 TCP 和 HTTP flood 是必备功能

统计发现每种 bot 都有集成了 3 种以上的 DDoS 功能，其中 TCP flood 和 HTTP flood 出现频率最高，有的 bot 甚至在此基础上发展出更多的攻击形式。

1.4 使用私有协议

所有 13 种 bot 的 C&C 协议都是私有协议，未发现使用标准的应用层协议（比如 HTTP）传输指令的情况。各种 bot 的 C&C 协议各不相同，但有一些共同点：

1. 运行时都有个类似注册的过程：bot 连接 C&C 服务器成功后会首先发送一个包含自身配置信息（比如 CPU 频率、操作系统版本等）的报文。

2. botmaster 可以在注册包中设置一些自定义值的实现版本和错误检测。比如 darkshell.c bot 的注册包里就有一个专门用于描述 bot 版本的字符串字段，观察发现不同的 darkshell.c bot 生成的注册包此字段明显不同，甚至即使同一个 botnet 在不同时期分发的 bot 样本，此字段也可能会变化。

3. 运行端口可自由设置。

1.5 样本普遍加壳

从加壳统计看 bot 样本在分发时普遍做了加壳保护，有极个别

的未加任何壳，但其 botnet 并不活跃，估计是处于试运行阶段，开发者还未考虑做加壳处理。

有一种 bot 的样本虽然未做加壳处理，但采取了加花机制，通过在样本中添加混淆指令或者混淆原来的执行流程来增加样本的逆向分析难度。

1.6 均留有后门

尽管主要是用于 DDoS 攻击，但分析发现这 13 种 bot 都保留了后门功能，botmaster 能控制僵尸主机下载并执行任意的可执行程序，实现远程安装。此外，不少 bot 都集成了远程关机、重启功能，超过一半的 bot 还集成了 C&C IP/port 更新功能。

后面将会提到，观察发现这些后门中使用最多的是下载 gh0st RAT 工具，让 botmaster 完全控制僵尸主机。

1.7 伴随各种 gh0st 变种

统计下载指令时发现 gh0st 是下载次数最多的一类软件，共检测到 644 个、41 种 gh0st 样本，所以有必要专门分析一下。

gh0st 本来是国内的一款开源 RAT 软件 [4]，用于远程计算机管理，其全面的功能加上源代码开放，使其深受黑产界青睐，常被改做木马使用，以至于出现了各种各样的 gh0st 变种 [5]。

gh0st 通信报文的特点是前面有 13 字节的报文头，里面包含一个特征串和两个长度字段，报文头紧跟的是经过 zlib 压缩的 payload，其压缩前、后的长度分别由报文头中的 2 个长度字段标识。gh0st 变种间的差别在报文中表现为：

1. 报文头结构不同：特征串可能在前 (gh0st1), 也可能位于报文头末尾 (gh0st2)。

2. 报文原始报文长度 (len2) 不同。

下面列出我们曾检测过的 gh0st 变种, 更多的变种信息可以参考 [5]。值得注意的是, 我们还发现了多个集成了 DDoS 攻击功能的 gh0st 变种, 相关的指令检测工作正在进行中。

```
gh0st1 -signature Black -len2 280
gh0st1 -signature ChEnA -len2 688
gh0st1 -signature Eyes1 -len2 1012
gh0st1 -signature Eyes2 -len2 932
gh0st1 -signature FKJP3 -len2 228
gh0st1 -signature Gh0st -len2 280
gh0st1 -signature Gh0st -len2 300
gh0st1 -signature Gh0st -len2 316
gh0st1 -signature Gh0st -len2 328
gh0st1 -signature Gh0st -len2 332
gh0st1 -signature Gh0st -len2 336
gh0st1 -signature Gh0st -len2 372
gh0st1 -signature Gh0st -len2 376
```

```
gh0st1 -signature Gh0st -len2 388
gh0st1 -signature Gh0st -len2 412
gh0st1 -signature Gh0st -len2 552
gh0st1 -signature Gh0st -len2 656
gh0st1 -signature Gh0st -len2 664
gh0st1 -signature HeiSe -len2 368
gh0st1 -signature HGChU -len2 720
gh0st1 -signature https -len2 284
gh0st1 -signature KrisR -len2 588
gh0st1 -signature Shado -len2 368
gh0st1 -signature Tyjhu -len2 416
gh0st1 -signature Winds -len2 364
gh0st1 -signature Winds -len2 664
gh0st1 -signature XIAOO -len2 228
gh0st1 -signature Xjihj -len2 332
gh0st1 -signature Xjihj -len2 412
gh0st1 -signature Xjihj -len2 632
gh0st1 -signature Xjihj -len2 636
gh0st1 -signature Xjihj -len2 712
gh0st1 -signature YinLe -len2 496
gh0st2 -len2 248
```

```
gh0st2 -len2 312
gh0st2 -len2 324
gh0st2 -len2 332
gh0st2 -len2 352
gh0st2 -len2 412
gh0st2 -len2 520
gh0st2 -len2 680
.....
```

二. C&C 服务器特点

在统计 botnet 时我们用 3 元组 (bot 类型、C&C 服务器、C&C 运行端口) 唯一标识一个 botnet, 所以 botnet 种类和 botnet 个数是一对多的关系。C&C 服务器统计主要从域名使用、地理分布、IP 地址解析等角度入手, 下面的描述反映了我们对检测到的 89 个 C&C 服务器的分析情况。

2.1 大多数 C&C 服务器分配了域名

在检测到的 89 个 C&C 服务器中, 只有 15 个没有域名, 其他均分配了域名, 这说明给 C&C 服务器分配域名在国内是主流。

反观欧美一些规模比较大的 botnet，往往会不分配 C&C 域名，而是直接使用纯 IP 连接。为何有这种差别，现在还不太清楚，需要继续观察。

从所使用的域名后缀看，3322.org 后缀的域名出现最多，达 20 个。值得一提的是 2012 年 9 月份微软为了打击 Nitel botnet “劫持”不少 3322.org 域名，那段时间我们一直观察的几个域名也在被劫持之列，域名都解析到了美国的 IP，所以那段时间内新出现的 gh0st bot 多使用 IP 连接 C&C 服务器，而且自那次事件以后，陆续出现了采用新后缀的 C&C 域名，我们观察到的非 3322.org 域名大部分自那以后出现。

2.2 运行端口非标准

统计发现 bot 种类和其运行端口不存在绑定关系，而且大部分 botnet 运行在大于 1024 的非标准端口上，不到 1/3 的 botnet 运行在标准端口上，但均与标准端口本来的使用意图无关。比如我们发现多个运行在 81 端口的 botnet，该端口本来分配给了 Kerberos 用于身份认证，但实际的 C&C 通信与 Kerberos 完全无关。

2.3 C&C 域名和 IP 的对应

分析域名解析情况时发现实际解析的 IP 数要大于 C&C 域名数，实际检测到的 C&C 域名加 C&C IP 共有 89 个，但先后检测到的 C&C 服务器 IP 有 212 个，这说明 C&C 域名和 IP 存在一对多的关系。

绝大部分域名都会固定地解析到有限的几个 IP 上，只有极

个别域名映射 IP 较多，比如有 1 个 C&C 域名在近 3 个月的时间内曾先后解析到 32 个不同的 IP。目前还不清楚这些 IP 确实为该 botmaster 所拥有还是使用了被黑的服务器，但我们确实发现过国外的 botnet 拿被黑的服务器充当 C&C 服务器的情况，先后检测到其 C&C IP 有 100 多个，而且地理上是全世界分布。

统计还发现存在单个 IP 对应多个域名的情况，这说明同一个 botmaster 可能运行了多个 botnet。

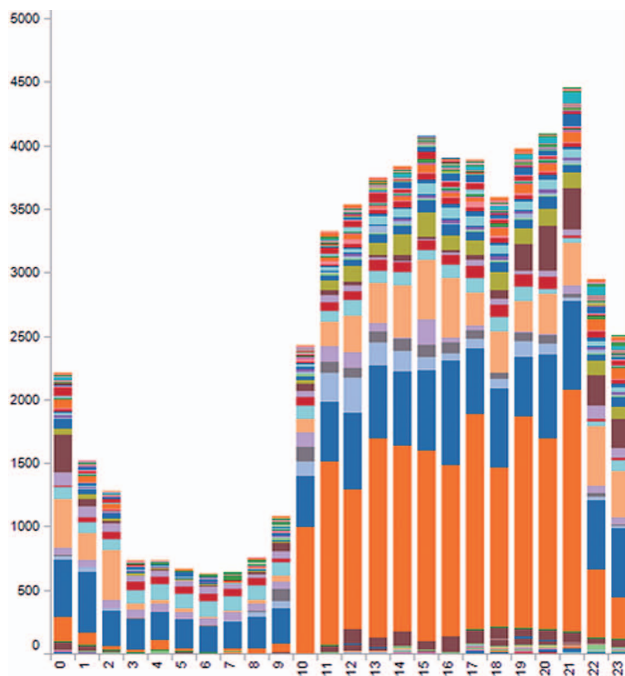


图 1 攻击指令 24 小时分布图

三. 运行和活跃情况

活跃情况主要依据攻击指令数和攻击频率来衡量, 我们半年来陆续检测到 27 万多条 DDoS 攻击指令, 被攻击目标有 17000 多个。如果将这些攻击指令按照 24 小时分布统计, 可以看出攻击主要发生在上午 10 点到晚上 12 点之间, 其中下午是最活跃的时段, 如图 1。

对被攻击目标分布统计表明, 被攻击目标主要位于大陆地区, 私服和电子商务网站出现比例最高, 也有个别政府、教育网站, 有意思的是攻击目标中多次出现了 DDoS “服务商” 网站。

从具体攻击参数看, 尽管每种 bot 都集成了多种类型的 DDoS 攻击功能, 但 botmaster 明显只偏爱某几种类型的攻击, 其中 TCP 和 HTTP flood 是使用最多的攻击手段。

从后门指令的使用看, 大部分 botnet 都检测到了后门指令, 而远程安装指令使用最多, 共检测到 11000 多条, 其他诸如关机、重启和更新 C&C 配置这些功能则极少使用, 不超过 100 条。

从统计看, 远程安装的内容集中在如下几种情况:

1. 安装 RAT 类软件, 最常见的是 gh0st 软件, 这个前面已经介绍。

猜测这么做的原因是为了完全控制僵尸主机。

2. 安装新的 bot 软件, 组建新的 botnet。

3. 将僵尸主机作为跳板, 继续攻击其他机器。

4. 安装一些刷流量的软件, 这种情况比较少见。

从后门的使用频率看 botmaster 非常希望完全控制僵尸主机, 以榨取更多的利益。另外, 某些 bot 尽管可以通过指令将已有 bot 迁

移到新的 botnet 中 (不同的 C&C 服务器 IP 和端口), 但 botmaster 更喜欢通过远程安装新 bot 的方式来实现迁移。

四. 总结

本文跟大家分享了我们观察到的大陆地区一些 DDoS botnet 的现象和特点, 实际中应该还有不少这类 botnet 未被观察到, 所以我们不能说这些特点对其他的都适用, 但管中窥豹, 我们相信在 botnet 的种类、运维方式、DDoS 攻击的手段等方面, 本文所总结的特点应该具有一定的代表性, 希望对大家有所帮助。

参考文献

[1] Over 9 million PCs infected - ZeroAccess botnet uncovered, <http://nakedsecurity.sophos.com/2012/09/19/zeroaccess-botnet-uncovered/>.

[2] Zeus Tracker, <https://zeustracker.abuse.ch/>.

[3] Win32/Gapz: New Bootkit Technique, <http://blog.eset.com/?p=16288>.

[4] Gh0st, 红狼安全小组, <http://www.wolfexp.net/>.

[5] The many faces of Gh0st Rat, http://www.norman.com/about_norman/press_center/news_archive/2012/the_many_faces_of_gh0st_rat/en/.

[6] Zeus (Trojan horse), http://enc.tfode.com/Gameover_%28trojan_horse%29.

Windows驱动常见漏洞分析

核心技术部 董阳

关键字：内核驱动 漏洞

摘要：本文主要讨论 Windows 驱动开发中特有的一些安全问题，在编写内核代码的过程中，如果缺乏相关的安全意识则很有可能留下一些潜在的安全漏洞，轻则造成系统崩溃，重则造成远程代码执行。本文还给出了若干建议，使开发人员在编写驱动程序时避免这些漏洞。

前言

近年来，随着 Windows 操作系统日益完善，多种内存保护技术的结合使得传统的基于缓冲区溢出的攻击越来越困难，在这种情况下，内核漏洞往往可以作为突破安全防线的切入点，例如 2011 年广为人知的 Duqu 病毒就是利用了 WIN32K 内核模块的 TrueType 字体解析引擎漏洞（MS11-087）。本文用四个例子讲解了部分在驱动编写过程中常见的驱动中的漏洞，最后提供了一些避免这些漏洞的建议。

例子 1：未使用 Probe 引发的漏洞

在内核模式下，由于各种原因程序员需要直接访问用户态的内存

```
1  VOID SetUserData (IN PCHAR DataPtr,IN ULONG DataLength)
2  {
3      if (DataLength > MAX_DATA_LENGTH)
4          DataLength = MAX_DATA_LENGTH;
5
6
7      memcpy(InternalStructure->UserData, pData, DataLength);
8      InternalStructure->UserDataLength = DataLength;
9  }
10
11  ULONG GetUserData (IN PCHAR DataPtr,IN ULONG DataLength)
12  {
13      if (DataLength > InternalStructure->UserDataLength)
14          DataLength = InternalStructure->UserDataLength;
15
16      memcpy (DataPtr, InternalStructure->UserData, DataLength);
17      return DataLength;
18  }
19
```

地址，经常使用用户态地址的场景包括 IOCTL/FSCTL、系统服务挂钩等等。在此过程中程序员如果操作不当，则可能引发较为严重的安全问题。

以上两个函数只是为了说明问题而构造的代码片段，通常出

现在 IOTCL 或者 FSCTL 这些地方。其中 SetUserData 函数接收一个用户态地址 DataPtr 并将数据复制到一个内核某地址处 InternalStructure->UserData。而 GetUserData 函数则正好相反，是将数据从内核某地址处 InternalStructure->UserData 复制到传入的用户态地址 DataPtr。

这个例子存在两种安全漏洞：

第一种是利用 SetUserData 函数把任意地址的数据写到 InternalStructure->UserData 里，然后通 GetUserData 函数把前面获取的内容读出来，从而造成读取任意地址内容可读的漏洞。

第二种是任意地址内容可写的漏洞。攻击者可以先利用 SetUserData 函数把要写入的数据写入 InternalStructure->UserData 里，再利用 GetUserData 把数据写入到任意地址。

造成以上漏洞的原因是，内核态函数在使用用户态指针的时候没有对该指针做有效性检查，因此攻击者可以通过传入一个有效的内核态地址从而达到对其进行读和写的操作。如果精心构造一些攻击数据甚至可以在

内核下执行 SHELLCODE。

例子 2：嵌入到 Buffered I/O 里的用户态指针

这里提供一个真实的漏洞，是之前在分析某安全产品驱动程序的时候发现的，这个漏洞是一个典型的在内核模式下利用嵌入到 Buffered I/O 里的用户态指针而没有对其进行检查的漏洞。该驱动的一个功能是通过 IOCTL 来将指定的文件内容读取到一个指定的缓冲区中，应用层通过一个结构体将应用层打开的文件句柄等相关参数传入到驱动层，利用驱动提供的接口读取目标文件。

```

if ( pFileReadInfo
    && InputBufferLength == 24
    && OutBufferLength == 24
    && ret
    && pFileReadInfo->FileHandle
    && pFileReadInfo->ReadBuffer )
{
    status = ObReferenceObjectByHandle(pFileReadInfo->FileHandle, 0x80000000u, 0, 0, &FileObj, 0);
    if ( status >= 0 )
    {
        ReadSize = pFileReadInfo->ReadSize;
        if ( ReadSize >= 0x400000 )
            ReadSize = 0x400000u;
        ReadOffset.LowPart = pFileReadInfo->OffsetLow;
        ReadOffset.HighPart = pFileReadInfo->OffsetHigh;
        pBuffer = AllocAndZero(0, ReadSize);
        FileBuffer = pBuffer;
        if ( pBuffer )
        {
            status = SendIrpReadFileData((PFILE_OBJECT)FileObj, pBuffer, ReadSize, (int)&ReadOffset, (int)&RetSize);
            if ( status >= 0 )
            {
                _RetSize = RetSize;
                if ( RetSize <= ReadSize )
                {
                    memcpy(pFileReadInfo->ReadBuffer, FileBuffer, RetSize); // 漏洞触发点
                    pFileReadInfo->ReturnSize = RetSize;
                }
            }
        }
    }
}

```

该 IOCTL 从应用层传入的结构体如下

所示：

```

//攻击时构造使用
typedef struct _READ_FILE_INFO
{
    HANDLE FileHandle;
    char* ReadBuffer;
    DWORD ReadSize;
    DWORD OffsetLow;
    DWORD OffsetHigh;
    DWORD ReturnSize;
}READ_FILE_INFO, *PREAD_FILE_INFO;

```

其中第一个参数 FileHandle 是目标文件的句柄，这个句柄通过应用层调用 CreateFile 函数得到，ReadBuffer 参数是目标缓冲区指针，ReadSize 是要读取数据的长度，OffsetLow 和 OffsetHigh 参数是要读取的文件偏移，ReturnSize 则是实际

读到的大小。

经过对驱动代码的分析发现，程序原本的意图是通过驱动向传入的文件句柄（文件对象）发送 IRP 请求来将文件的内容读取到指定的用户态缓冲区中，但是由于程序员在使用 ReadBuffer 的时候仅仅判断该指针是否为 NULL，没有判断该指针是不是一个合法的用户态指针，因此攻击者可以构造一个 READ_FILE_INFO 结构传入到驱动中，其中将 ReadBuffer 设置成除 0 地址以外的任意地址（包括内核地址），并构造一个特殊内容的文件（比如文件大小为 4 字节，内容为 0），让驱动读取到指定的地址。由此可以造成除 0 以外任意地址写入长度最大为 0x400000 字节内容的漏洞，利用较为流行的“用 0 覆盖 HalQuerySystemInformation”的技术就可以以 ring0 权限执行 SHELLCODE。

该驱动本身对 IOCTL 的调用者做了限制，判断了调用者文件的数字签名，因此直接去打开该驱动操作 IO 的话会失败，但是软件本身并没有限制句柄的复制，因此我们在软件自身打开驱动后将其获得的设备驱动句柄复制出来即可对该驱动进行操作。

对于上述问题的解决方案是：

1. 检查嵌入到 SystemBuffer 里的缓冲区的长度。
2. 利用 ProbeForWrite 函数对嵌入到 SystemBuffer 里的指针进行检查。
3. 在所有引用该指针的地方都使用 SEH。
4. 验证调用者是否合法，并且防止外部进程操作合法调用者的句柄（防注入、防句柄复制等）。

例子 2 续：驱动层使用用户态句柄的问题

上例中的代码其实还有另外一些问题，即在调用 ObReferenceObjectByHandle 函数的时候 ObjectType 参数使用了 NULL 参数，我们查看 ObReferenceObjectByHandle 函数的代码时可以发现（下图），当 ObjectType 参数设置为 NULL 的时候，可以返回任意类型的有效对象。如果在调用该 IOCTL 的时候传入的 FileHandle 为非文件类型的句柄时，当 ObReferenceObjectByHandle 函数调用完成后会返回一个有效的非文件对象，后续的代码会对该对象进行 IRP 相关的操作，从而造成系统崩溃，如果在特定条件下有可能执行内核 SHELLCODE。

```
if ((ObjectHeader->Type == ObjectType) || (ObjectType ==
NULL)) {
....
if ((SeComputeDeniedAccesses(GrantedAccess,
DesiredAccess) || (AccessMode == KernelMode)) {
....
```

此外如果 AccessMode 设置为 KernelMode 则会造成绕过对句柄访问权限的检查。例如用户传入的是一个只有读权限的句柄，但是调用了有对此句柄进行写操作的接口，则仍然会写文件成功。

因此，上述的代码应按照下面的方式调用：

```
status = ObReferenceObjectByHandle(pFileReadInfo->FileHandle, 0x80000000, *IoFileObjectType, UserMode, &FileObj, 0);
```

例子 3：使用 METHOD_NEITHER 方法向 IOCTLs 传参

METHOD_NEITHER 方法是直接将用户态的缓冲区指针传入到驱动当中，输入缓冲区在 IO_STACK_LOCATION 的

```

1 case IOCTL_GET_HANDLER: {
2     PULONG_PTR Value;
3     ULONG Data = 0x12345678;
4
5     Value = Irp->UserBuffer;
6     if Parameters.DeviceIoControl.OutputBufferLength >= sizeof(ULONG_PTR) {
7
8         try {
9             if (Irp->RequestorMode != KernelMode) {
10                ProbeForWrite(Value,
11                    sizeof( ULONG_PTR ),
12                    TYPE_ALIGNMENT( ULONG_PTR ));
13            }
14
15            *Value = (ULONG_PTR)Data;
16        } except( EXCEPTION_EXECUTE_HANDLER ) {
17        }
18        ...
19    }
20
21    if (Length != 0) {
22        if (((ULONG_PTR)Address & (Alignment - 1)) != 0) {
23            ExRaiseDatatypeMisalignment();
24        } else if (((ULONG_PTR)Address + Length) > (ULONG_PTR)MM_USER_PROBE_ADDRESS) ||
25                (((ULONG_PTR)Address + Length) < (ULONG_PTR)Address) {
26            *(volatile UCHAR * const)MM_USER_PROBE_ADDRESS = 0;
27        }
28    }
29 }

```

```

1 case IOCTL_GET_HANDLER: {
2     PULONG_PTR Value;
3     ULONG Data = 0x12345678;
4
5     Value = Irp->UserBuffer;
6     *Value = (ULONG_PTR);
7     ...
8 }

```

Type3InputBuffer 字段当中，而输出缓冲区在 IRP 的 UserBuffer 字段当中。当处理 METHOD_NEITHER 请求的时候，IO 管理器并不会去检查传入的缓冲区指针及其长度，因此驱动必须自己检查这些指针和长度以及对齐粒度，并在使用到该指针的地方使用 SEH 包裹。

通过上图中的例子可以看到，代码在没有对 Irp->UserBuffer 做任何检查的情况下就直接对其指针进行赋值，因此如果是攻击者则可以从用户层传入任意地址，此 IOCTL 被调用后造成任意地址写特定值的漏洞（在本例中是任意地址被写为 0x12345678），如果精心利用则可以执行内核 SHELLCODE 执行提权等操作。因此正确的使用方法如图所示：

解决方案是：

1. 检查缓冲区的长度。
2. 利用 ProbeForWrite 函数对指针进行检查。

```

// Attacker controls OutputBuffer and OutputBufferLength
void IOCTL_handler(...)
{
    [...]
    try {
        ProbeForWrite (OutputBuffer,
            OutputBufferLength, // [1] length could be zero and this will be bypassed even with a kernel mode
            address
            sizeof (UCHAR));

        RtlCopyMemory(OutputBuffer,
            (PUCHAR)context->endpoint->Common.VcConnecting.RemoteSocketAddressOffset,
            endpoint->Common.VcConnecting.RemoteSocketAddressLength // [2] copy based on a non-zero
            value to a kernel mode address
            );

    }except(AFD_EXCEPTION_FILTER(&status)){} }
    [...]
}

```

3. 在所有引用该指针的地方都使用 SEH。

例子 4：ProbeByPass

我们先来看看 ProbeForRead 函数的代码：

可以看到，ProbeForRead 函数首先判断数据的长度是否为 0。如果数据长度不为 0，则判断传入的地址 Address 是否按设置的 Alignment 参数对齐。如果数据没有按照指定的对齐参数对齐，则会调用 ExRaiseDatatypeMisalignment() 函数从而触发异常。因此只要传入的地址跟对齐粒度不匹配即可触发异常。这种情况下有可能会绕过某些主动防御软件的过滤拦截，这个问题早在几年前 MJ0011 的博客就有过论述。

而我们要讨论的另一个问题是，如果传入的缓冲区长度是 0，则会绕过 ProbeForXXX 的检查，从而导致对非期望的地址进行读写。这里比较典型的就是 MS08-66 这个漏洞了。我们先看下这里的代码，MS08-066 是 AFD.SYS 模块造成的安全漏洞，AFD.SYS 模块是系统用于 SOCKET 通信的内核驱动程序：

可以由上图看到，代码在 [1] 处使用了 ProbeByWrite 对传入的 OutBuffer 进行检查，并且在所有引用处都使用了 SEH 进行包裹。乍一看似乎没有什么问题，但实际上，如果用户态调用者传进来的 OutputBufferLength 为 0 的话，则会造成绕过 ProbeForWrite 函数的检查，从而导致 [2] 处的代码直接向应用层调用者设置的地址（包括内核地址）进行数据拷贝，从而导致任意地址可写的漏洞。

总结

为了避免以上提到的若干类漏洞，驱动开发人员在编写相关代码的时候应遵守以下规则：

- 1、在内核模式下使用用户态指针前一定先用 ProbeXXX/MmProbeAndLockPages 函数对指针进行测试，确定目标内存是否可读 / 写。
- 2、所有引用用户态指针的地方都要使用 try...except 包裹，而不仅仅是使用 ProbeXXX 的地方。
- 3、要假设用户态指针可以按任意值对齐，从而避免因对齐粒度不符而造成的异常，这种问题可能会绕过某些安全软件的过滤拦截。
- 4、用户态内存可能随时变化，比如被其他线程所修改，因此不要使用传入的用户态内存作为临时存储空间，也不要认为两次从一个用户态内存里获取到的数据就一定相同。
- 5、检查所有从用户态代码传入的数据。
- 6、检查传入的缓冲区长度。
- 7、正确设置 ObReferenceObjectByHandle 函数的参数类型 (AccessMode, ObjectType)。

参考文献

- 1、《Common Driver Reliability Issue》Microsoft Corporation
- 2、<http://hi.baidu.com/mj0011/item/e57573340dc583302e0f81f0>
- 3、Windows 系统中错误的句柄引用所引发的漏洞 (http://www.whitecell.org/file/wss_paper.rar)

Z4root剖析

安全研究部 刘永军

关键词：Z4root 工具 实现

摘要：Z4root 是非常流行的一款获取 Android 2.3 之前版本系统 root 权限的工具，本文简要介绍其使用，详细阐述其实现的技术细节。

Root 是什么

Android 系统出于安全原因考虑，默认程序是以非 root 用户权限运行的，这就导致用户不能随心所欲地删除厂商强制安装的程序，定制个性化系统，而且使那些需要 root 权限才能正常运行的功能强大的软件（如 Root Explorer、杀毒软件等）功能受限。“无 Root，不 Android”！root，是你成为“Android 控”的必要条件。

Root，即获取 Android 系统 root 权限，root 是 Android 系统中的超级管理员用户账户，这个管理员的权限非常高，基本上可以操作手机中的一切数据，相当于 Windows 的 administrator 权限。

Root 最终目的简单讲就是往系统 /

system/bin（也可以是其他 path 变量路径，只是方便调用）下放置一个具有 SUID 标志位的修改版 su 可执行程序，其他程序如果需要 root 权限只需要调用 su 即可。为了便于授权管理，还需要安装一个 superuser.apk 管理程序。

实现上述目的可以通过刷第三方刷机包、fastboot 等方式，但均有其局限性，如可能需要解锁或者 fastboot 方式不被手机厂商支持等。最常用的是利用 Android 系统漏洞获取 root 权限后实现 root 目的。Z4root 就是利用 Android 2.3 之前版本 setuid 漏洞实现 root 的工具。

Z4root 简介

Z4root 是来自 XDA 的 RyanZA 发布的，

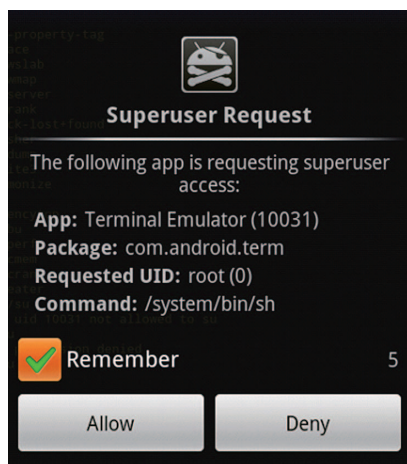
其主界面如下图：



主界面上面有三个选项，即“Temporary（临时）Root”、“Permanent（永久）Root”、“Un-root”，运行 Z4root 需要打开“USB 调试”，否则 Z4root 不能够进行 root 操作。

Root 成功后，当有程序请求 root 权限

时，授权管理对话框弹出，如下图：



Z4root 在模拟器 (Android 2.2) 中运行会与真机有一些差异，如下：

1) 由于模拟器中默认存在 `/system/xbin/su`，所以即使第一次安装 Z4root 按钮“un-root”也有效，并提示 su 存在，但其为系统默认 su，不能完成为其他程序授予 root 权限功能。

2) 启动模拟器需要加 `-partition-size`，否则会由于 `system` 分区空间不足而导致临时 root 失败。而且 `system` 是只读的，需要临时 root 前调用 `adb remount` 去掉只读属性。

3) 永久 root 会失败，因为永久 root 后

自动重启系统模拟器会卡住，手动重启后 `system` 分区会自动恢复，导致永久 root 失败。

setuid 漏洞介绍

setuid 用来设置实际用户 UID 和有效用户 UID，其在目标用户进程数达到 `RLIMIT_NPROC` 极限时返回错误，这一问题可能产生的安全隐患最早可以追溯到 2000 年，而在 2006 年，出现了真正利用这一编码问题的漏洞 (CVE-2006-2607)。

Android 2.3 之前版本 `adb` 程序 (adb 服务端) 存在 `setuid` 漏洞，`src/system/core/adb/adb.c` 中漏洞代码如下：

```
setuid(AID_SHELL);
```

如果 `adb` 为 root 进程，调用 `setuid` 将用户从 root 切换回 shell，但在 shell 用户进程数达到上限 `RLIMIT_NPROC` 时会失败，因此进程继续以 root 身份运行，而没有返回。2.3 及以后版本修改如下：

```
if (setuid(AID_SHELL) != 0) {
    exit(1);
}
```

Androidadbsetuid 漏洞的利用首先要

以 shell 用户身份创建大量僵尸进程，当进程总数达到 shell 用户进程数上限 (`RLIMIT_NPROC`) 时，杀掉 `adb` 进程 (以 SHELL 用户身份运行)，这时系统中 SHELL 用户运行的总进程数为 (`RLIMIT_NPROC-1`)，所以需要立刻创建一个新进程使得 shell 用户总进程数仍然是 `RLIMIT_NPROC`。等系统发现 `adb` 不再运行时重新启动一个新进程，该进程最初以 root 身份运行，然后会调用 `setuid()` 切换至 shell 用户身份，但此时 shell 用户的进程数已经达到上限，所以 `setuid` 操作失败，用户切换没有成功，`adb` 还是 root 权限。程序继续往后执行，并最终产生了一个具有 root 权限的 `adb` 进程，用户可以利用这个 `adb shell` 执行 root 操作。

`rageagainstthecage` 是互联网上很容易搜索到的一个 exploit 程序，可以通过 `adb shell` 调用此程序获得 root shell，进而实现 root 目的。因为打开“USB 调试”才能有 `adb` 进程，所以需要手机打开“USB 调试”功能，`rageagainstthecage` 检测到 `adb` 进程存在后才会正常执行。

Z4root 由于使用了 `rageagainstthecage`

程序，所以同样需要手机打开“USB 调试”功能，这很容易给人错觉以为其利用的是 adbsetuid 漏洞，其实不然。我们在研究过程中发现 Z4root 实现 root 本质上利用的是另外一个 zygote setuid 漏洞，故打开“USB 调试”也不是必须的。

Z4root 具体实现分析

1) 利用漏洞获取 root 权限进程

上述 rageagainstthecage 利用程序创建了大量 shell 用户进程，使 shell 用户进程数达到上限，进而利用 adbsetuid 漏洞获取 root 权限，但 Z4root 为 app 用户，其调用 rageagainstthecage 创建的大量用户也为 app 用户，所以 Z4root 应该不能成功利用 adbsetuid 漏洞，那它是如何 root 成功的呢？

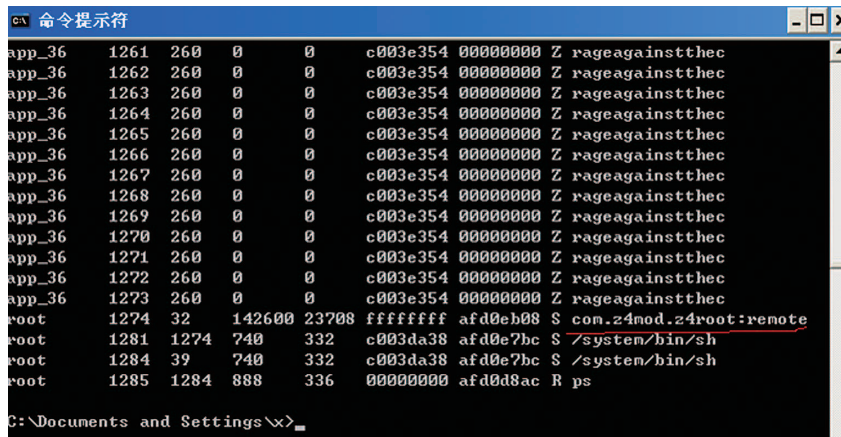
Z4root 通过 rageagainstthecage 程序 fork 大量 Z4root 用户进程，达到其用户进程数上限，然后通过延时触发广播能够创建一个具有 root 权限的 com.z4mod.z4root:remote 进程，其广播接收器在 AndroidManifest.xml 中声明如下：

```
<receiver :process=".remote"android:
```

```
name="AlarmReceiver">
```

```
</receiver>.
```

进程创建结果如下图红色标线：



```
命令提示符
app_36 1261 260 0 0 c003e354 00000000 Z rageagainstthec
app_36 1262 260 0 0 c003e354 00000000 Z rageagainstthec
app_36 1263 260 0 0 c003e354 00000000 Z rageagainstthec
app_36 1264 260 0 0 c003e354 00000000 Z rageagainstthec
app_36 1265 260 0 0 c003e354 00000000 Z rageagainstthec
app_36 1266 260 0 0 c003e354 00000000 Z rageagainstthec
app_36 1267 260 0 0 c003e354 00000000 Z rageagainstthec
app_36 1268 260 0 0 c003e354 00000000 Z rageagainstthec
app_36 1269 260 0 0 c003e354 00000000 Z rageagainstthec
app_36 1270 260 0 0 c003e354 00000000 Z rageagainstthec
app_36 1271 260 0 0 c003e354 00000000 Z rageagainstthec
app_36 1272 260 0 0 c003e354 00000000 Z rageagainstthec
app_36 1273 260 0 0 c003e354 00000000 Z rageagainstthec
root 1274 32 142600 23708 ffffffff afd0eb08 S com.z4mod.z4root:remote
root 1281 1274 740 332 c003da38 afd0e7bc S /system/bin/sh
root 1284 39 740 332 c003da38 afd0e7bc S /system/bin/sh
root 1285 1284 888 336 00000000 afd0d8ac R ps
C:\Documents and Settings\>
```

为何能够创建出 root 权限的 com.z4mod.z4root:remote 进程？具体原因如下：

创建应用进程由 root 用户进程 zygote 调用 src\dalvik\vm\native\dalvik_system_Zygote.c 中 forkAndSpecializeCommon 函数实现，

Android2.1 forkAndSpecializeCommon 函数漏洞部分如下：

```
err = setgid(gid);
if (err < 0) {
    LOGW("cannot setgid(%d) errno: %d", gid, errno);
}

err = setuid(uid);
if (err < 0) {
    LOGW("cannot setuid(%d) errno: %d", uid, errno);
}
```

▶▶ 前沿技术

此时 uid 为 Z4root 用户 uid，因为已达进程数上限，setuid 失败，但并没有返回，而是继续向下执行，进而导致创建的进程以 zygote 相同的 root 用户权限运行。下图为 Android 2.3.7 (2.3 及以后版本已经修补) 漏洞修补后的代码：

```
err = setgid(gid);
if (err < 0) {
    LOGE("cannot setgid(%d): %s", gid, strerror(errno));
    dvmAbort();
}

err = setuid(uid);
if (err < 0) {
    LOGE("cannot setuid(%d): %s", uid, strerror(errno));
    dvmAbort();
}
```

综上所述，Z4root 之所能够 root 成功，是其也许“无心插柳”地利用了另外一个 zygote setuid 漏洞。既然不是利用的 adbsetuid 漏洞，打开“USB 调试”也就不是必须的了。

2) 功能具体实现

通过上述获取 root 权限进程后，利用 busybox 工具集复制 su 等文件到相应目录，安装 superuser.apk 等。busybox、su、superuser.apk、rageagainstthecage 程序包含在 Z4root 的 res/raw 目录下，前三者为压缩文件。

Z4root 功能菜单分析如下：

a) 临时 root

主要实现代码如下图：

Kill 掉所有 rageagainstthecage 进程，创建 ext2 格式设备，备份 system/bin 目录下内容，mount 设备到 system/bin 目录，恢

```
write(out, "chmod 777 " + getFilesDir() + "/busybox");
write(out, getFilesDir() + "/busybox Killall rageagainstthecage");
write(out, getFilesDir() + "/busybox Killall rageagainstthecage");
write(out, getFilesDir() + "/busybox rm " + getFilesDir() + "/tmproot.ext");
write(out, getFilesDir() + "/busybox rm -rf " + getFilesDir() + "/bin");
write(out, getFilesDir() + "/busybox cp -rp /system/bin " + getFilesDir());
write(out, getFilesDir() + "/busybox dd if=/dev/zero of=" + getFilesDir() + "/tmproot.");
write(out, getFilesDir() + "/busybox mknod /dev/loop9 b 7 9");
write(out, getFilesDir() + "/busybox losetup /dev/loop9 " + getFilesDir() + "/tmproot.");
write(out, getFilesDir() + "/busybox mkfs.ext2 /dev/loop9");
write(out, getFilesDir() + "/busybox mount -t ext2 /dev/loop9 /system/bin");
write(out, getFilesDir() + "/busybox cp -rp " + getFilesDir() + "/bin/* /system/bin/");
write(out, getFilesDir() + "/busybox cp " + getFilesDir() + "/su /system/bin/");
write(out, getFilesDir() + "/busybox cp " + getFilesDir() + "/busybox /system/bin/");
write(out, getFilesDir() + "/busybox chown 0 /system/bin/su");
write(out, getFilesDir() + "/busybox chown 0 /system/bin/busybox");
write(out, getFilesDir() + "/busybox chmod 4755 /system/bin/su");
write(out, getFilesDir() + "/busybox chmod 755 /system/bin/busybox");
write(out, "pm install " + getFilesDir() + "/SuperUser.apk");
```

复 system/bin 目录下所有内容，复制 su、busybox 到 system/bin，安装 superuser.apk。

b) 永久 root

主要实现代码如下图：

```
command = "chmod 777 " + getFilesDir() + "/busybox\n";
out.write(command.getBytes());
out.flush();
command = getFilesDir() + "/busybox mount -o remount,rw /system\n";
out.write(command.getBytes());
out.flush();
command = getFilesDir() + "/busybox cp " + getFilesDir() + "/su /system/bin/\n";
out.write(command.getBytes());
out.flush();
command = getFilesDir() + "/busybox cp " + getFilesDir() + "/SuperUser.apk /system/app/\n";
out.write(command.getBytes());
out.flush();
command = getFilesDir() + "/busybox cp " + getFilesDir() + "/busybox /system/bin/\n";
out.write(command.getBytes());
out.flush();
command = "chown root.root /system/bin/busybox\nchmod 755 /system/bin/busybox\n";
out.write(command.getBytes());
out.flush();
command = "chown root.root /system/bin/su\n";
out.write(command.getBytes());
out.flush();
command = getFilesDir() + "/busybox chmod 4755 /system/bin/su\n";
out.write(command.getBytes());
out.flush();
command = "chown root.root /system/app/SuperUser.apk\nchmod 755 /system/app/SuperUser.";
out.write(command.getBytes());
out.flush();
command = "rm " + getFilesDir() + "/busybox\n";
out.write(command.getBytes());
out.flush();
command = "rm " + getFilesDir() + "/su\n";
out.write(command.getBytes());
out.flush();
command = "rm " + getFilesDir() + "/SuperUser.apk\n";
out.write(command.getBytes());
out.flush();
```

修改 system 分区只读属性，复制 su、busybox 到 /system/bin，复制 superuser.apk 到 /system/app 实现 superuser 安装，修改文件所有者、属性，删除临时文件，最后重启系统完成永久 root。

c) 清除 root

主要实现代码如下图：

```
write(out, "chmod 777 " + getFilesDir() + "/busybox");
write(out, getFilesDir() + "/busybox mount -o remount,rw /system");
write(out, getFilesDir() + "/busybox rm /system/bin/su");
write(out, getFilesDir() + "/busybox rm /system/sbin/su");
write(out, getFilesDir() + "/busybox rm /system/bin/busybox");
write(out, getFilesDir() + "/busybox rm /system/sbin/busybox");
write(out, getFilesDir() + "/busybox rm /system/app/SuperUser.apk");
write(out, "echo \"reboot now!\"");
saystuff("Rebooting...");
Thread.sleep(3000);
write(out, "sync\nsync");
write(out, "reboot");
```

修改 system 分区只读属性，删除 /system/bin、/system/sbin 下的 su、busybox，删除 /system/app 下 superuser.apk 实现卸载 superuser，重启系统完成 root 清除。

3) su 和 superuser

su 检索数据库 (/data/data/com.koushikdutta.superuser/databases/superuser.sqlite)，如果找到记录的话，说明当前进程已经被用户允许获取 root 权限，则改变当前进程的用户 ID 和组 ID。代码如下：

```
if(setgid(gid) || setuid(uid))
    returnpermissionDenied();
```

如果没有检索到的话，su 会通过 am start 命令打开 superuser.apk 中的

SuperuserRequestActivity。代码如下：

```
if (!checkWhitelist())
{
    charsysCmd[1024];
    sprintf(sysCmd, "am
start -a android.intent.action.MAIN -n com.
koushikdutta.superuser/com.koushikdutta.
superuser.SuperuserRequestActivity --eiuid
%d --eipid %d > /dev/null", g_puid, ppid);
    if (system(sysCmd))
returnexecutionFailure("am.");
```

上述为旧版本 su 程序的主要处理流程，新版本 su 处理流程与旧版类似，主要在 su 与 superuser 通信机制方面有所变化：通过更底层的 activity 服务调用 superuser 中的 activity；判断 superuser 管理程序用户的选择操作不像旧版通过轮询数据库的方式，而采用了 socket 即时通信的方式。而且新版 su 判断如果是 shell 用户则直接赋予 root 权限，无需用户参与。

Superuser.apk 为授权管理程序，可以查看授权日志及管理已授权程序等，无需赘述。

个人网上银行登录安全研究报告提要

北京分公司 赵波 李哲祎 刘凯

摘要：个人网上银行登录安全研究是以 2012 年 10 月 24 日标准普尔在北京发布的《中国 50 大银行》报告中提及的 50 大银行为对象，对这 50 大银行的个人网上银行登录进行了调查、分析和深入研究，借此来巩固并完善在“入口安全”领域的安全积累。

此项研究我们分别从现状、攻与防、监管、局限性四个方面进行探讨。通过研究我们了解了当前中国 50 大银行个人网上银行业务中的登录安全现状，掌握了当前与登录安全相关的第一手宝贵资料。同时，也希望借助此项研究，总结经验，将其反馈给中国各大银行，为中国银行业网上银行的信息安全事业尽一份微薄之力，为推动中国银行业网上银行整体安全持续、健康、稳定的发展做出一份贡献。

现状：安全措施日益多样，且细节丰富

网上银行普遍采用的安全措施包括安全会话、身份鉴别、输入保护、验证码和失败处理等等，这些安全措施都是经过一系列的安全事件的洗礼逐步建立起来的。各大银行在针对上述安全措施的运用上存在较大差异。是否使用了安全措施固然重要，但是否真正发挥了安全措施的作用才是根本。

1) 安全的 SSL 会话是网上银行登录的第一道防线

基于对 50 大银行登录会话安全的调查数据和分析，结果显示：被调研的所有个人网上银行的网络通讯均采用 HTTPS 方式来确保

建立一个安全的信道。

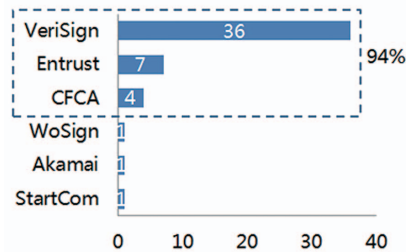
研究显示：国内银行更容易接受 VeriSign 颁发的 SSL 证书，除 VeriSign 外，Entrust、CFCA 也是常见的证书颁发机构（见图 D.1.1）。

2) 多因素认证的身份鉴别方式已经得到普及

研究显示：登录过程中进行多因素认证的个人网上银行已经基本普及，约占 82%；采用 USBKEY 证书认证的方式最为常见，达到 79%（见图 D.1.2、图 D.1.3）。

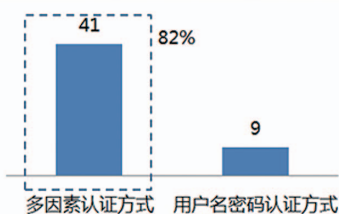
同时，银行依然愿意采用传统的用户名密码认证，并且不仅仅

D.1.1 对安全的SSL会话的认知
VeriSign是最容易接受的SSL证书颁发机构



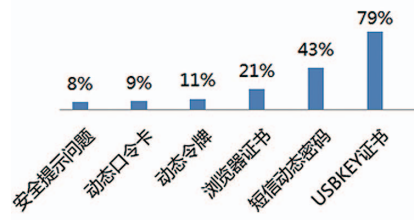
备注：数据样本为中国50大银行
来源：个人网上银行登录安全研究（2013年1月），绿盟科技

D.1.2 身份鉴别的两种常见方式
多因素的身份鉴别比传统用户名密码方式更普及



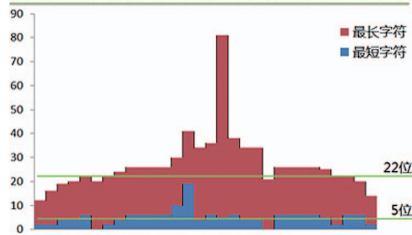
备注：数据样本为中国50大银行
来源：个人网上银行登录安全研究（2013年1月），绿盟科技

D.1.3 多种多因素认证方式的使用情况比较
采用USBKEY证书的多因素认证登录方式最为常见



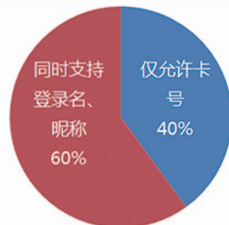
备注：由于各银行个人网银并非支持一种认证方式，因此各种认证方式百分比之和并非百分之百
来源：个人网上银行登录安全研究（2013年1月），绿盟科技

D.1.5 用户名和密码的安全策略要求更加严格
登录用户名长度的正态分布显示其长度策略主要集中在5-22位字符



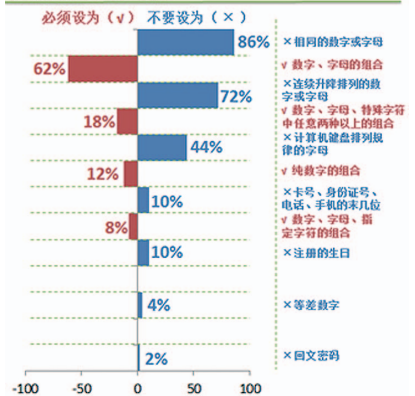
备注：仅针对研究发现的明确要求用户名长度的30家个人网上银行进行统计
来源：个人网上银行登录安全研究（2013年1月），绿盟科技

D.1.4 用户名密码认证方式的使用情况比较
五分之三个人网上银行支持用户自定义登录名、昵称



备注：数据样本为中国50大银行
资料来源：个人网上银行登录安全研究（2013年1月），绿盟科技

D.1.6 密码复杂度策略是6-8位短密码实现强鲁棒性的必要保证



备注：1) 数据样本为中国50大银行；2) 表格是健壮和强性的意思；3) 回文密码指从左向右或从右向左完全相同的密码。
来源：个人网上银行登录安全研究（2013年1月），绿盟科技

只能通过卡号登录、而是允许用户自定义便于记忆的个性登录名或昵称（见图 D.1.4），但对登录名或昵称长度、密码长度、密码复杂度都有严格的安全策略要求（见图 D.1.5、图 D.1.6）。

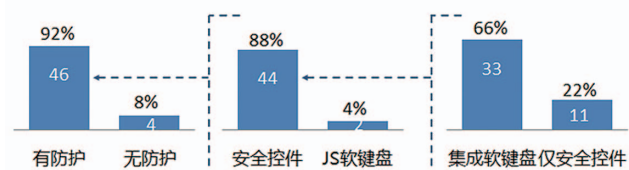
3) 对密码的输入保护多采用安全控件集成软键盘的混合模式

登录过程中，密码是保护对象中安全级别要求最高的。

研究显示：当前输入保护主要有“安全控件”和“软键盘”两种手段，但当前的部分安全控件已经得到创新，将软键盘的功能集成到控件之中，这种混合模式的安全控件也常被多数网上银行采用（见图 D.1.7）。

D.1.7

安全控件集成软键盘模式是对密码输入保护的主流方式



备注：JS软键盘是指通过JavaScript技术实现的Web前端软键盘

来源：个人网上银行登录安全研究（2013年1月），绿盟科技

4) 验证码依然是登录中不可避免的负面体验

在研究中发现 90% 的个人网上银行登录均需要输入验证码，验证码给用户的个人体验依然是负面的——平均延迟用户登录约 2 秒钟。

针对验证码的使用情况调研结果请见图 D.1.8。

5) 登录失败策略差异较大，但反馈信息多使用“友好的错误提示”，并进行了模糊化处理

“失败处理”是用户登录个人网上银行失败时，网上银行系统对登录失败采取的安全保护策略，而失败反馈给用户的各种提示信息，称为“错误提示”。

研究显示：登录失败处理的安全保护策略近 94% 的网上银行采用“账号自动锁定策略”，即达到特定次数的失败登录，账号会自动锁定特定时间，而满足特定时间后，账号再自动解锁。

6) 少数银行对客户端的浏览器要求更精细

“浏览器功能屏蔽”是银行为防范个人网上银行客户端风险，采取的屏蔽客户端浏览器部分功能的措施一般包括屏蔽菜单栏功能、屏蔽导航栏功能、屏蔽右键功能等。

7) 预留信息的作用并不明显

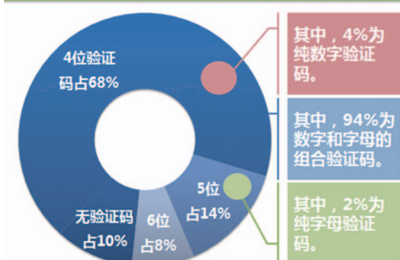
预留信息是银行为帮助用户有效识别个人网上银行、防范不法分子利用假银行网站进行网上诈骗的一项安全措施。

研究显示：当前个人网上银行采用的预留信息方式主要以文字防伪标识为主，少数银行也提供图片防伪标识的预留信息（见图 D.1.12）。

8) 欢迎首页的登录提醒信息多种多样

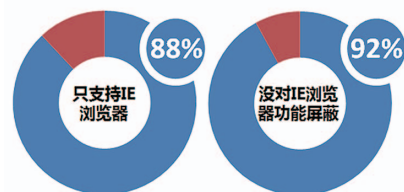
当成功登录个人网上银行后，登录页面提示的用户个人信息或以往的登录历史信息，称其为“登录提醒信息”。

D.1.8 对登录验证码的认知 4位数字和字母的组合验证码最为常见



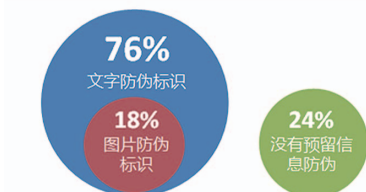
备注：数据样本为中国50大银行
来源：个人网上银行登录安全研究（2013年1月），绿盟科技

D.1.11 多数银行对客户端的浏览器关注不够



备注：数据样本为中国50大银行
来源：个人网上银行登录安全研究（2013年1月），绿盟科技

D.1.12 76%网上银行登录时采用的预留信息是文字标识



备注：数据样本为中国50大银行
来源：个人网上银行登录安全研究（2013年1月），绿盟科技

研究显示：当前各网上银行采取的登录提醒信息多种多样，较为多见的是提醒“上次登录的时间”信息（见图 D.1.13）。

9) 与登录相关的限制策略应用百花齐放、百家争鸣，但效果不佳

在网上银行登录过程中，除登录事件中可见的各种安全措施外，还有一些非常隐性的措施，以防范各种未知的登录风险，称其为“与登录相关的限制策略”，也可理解为“与登录相关的安全加强策略”。

研究显示：限制策略主要体现在四种，分别为“密码有效期限制”、“登录电脑限制”、“登录时间或IP地址限制”和“登录频率或地域限制”。只有11家银行的个人网上银行具备一项或多项可设置限制策略的功能，且遗憾的是这些功能中多数为被动设置项，而非默认设置项。

攻与防：解决最突出的五大威胁是保障网上银行登录安全的关键

在攻击方面，当前对网上银行最突出的威胁有网络钓鱼、恶意代码、暴力破解、恶意滥用、身份假冒五种。其中，前三种是针对网上银行提供者的，即银行；而后两者是针对网上银行用户的，即用户。

在防护方面，对抗五大威胁的手段主要有防钓鱼、防数据泄露、防猜测、防滥用、防盗用五类措施。

10) 网络钓鱼是个人网上银行登录最大的安全威胁

网络钓鱼威胁会直接带来经济损失。从防御角度银行设计了“预留信息”的功能和采用扩展验证(EV)SSL证书进行可信提醒，这些防御措施都有一定的局限性。

通过调研与分析认为：银行方面应加强网上银行用户者安全意识的宣贯工作，促使用户养成良好的安全使用习惯，才能更好地发挥网上银行安全措施的作用。

D.1.13

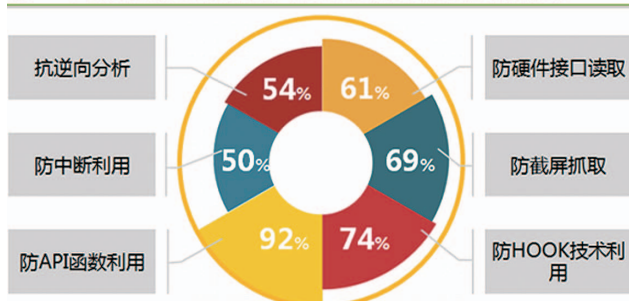
80%的网上银行登录提醒信息是“上次登录时间”



备注：1) 数据样本为中国50大银行。2) 图中N为自然数列。
来源：个人网上银行登录安全研究（2013年1月），绿盟科技

D.2.2

防API函数利用技术效果较好，防中断和抗逆向能力有待提升



备注：1) 数据样本为中国50大银行。2) 百分数采取四舍五入取整。
来源：个人网上银行登录安全研究（2013年1月），绿盟科技

11) 恶意代码攻击与网上银行登录紧密相关

恶意代码攻击从未停止脚步，攻击的战场也涉及到了网上银行领域。

通过调研与分析发现：在采用安全控件的网上银行中，多数安全控件能够防御常见的截获攻击，如 Windows API 截获、HOOK 截获、键盘中断等，并且近半数的安全控件具备抗逆向分析的能力。另外，在使用软键盘的网上银行中，近半数软键盘进行了防截屏设计（见图 D.2.2）。

12) 暴力破解攻击在逐步降低，但仍不可忽视

暴力破解攻击一度风靡，利用自动化的软件，可获得大量使用较弱口令的登录账户。网上银行在进行登录暴力破解防御方面主要采用限制策略和验证码技术两类安全措施。

通过调研与分析认为：由于限制策略和验证码的使用，使暴力破解攻击成功率大大降低，在成本远大于利益的现实面前，暴力破解攻击事件正在逐步降低。同时，由于用户的安全意识薄弱，少数银行对密码的要求不是十分严格，也造成了暴力破解攻击一直持续不断，仍不可忽视。

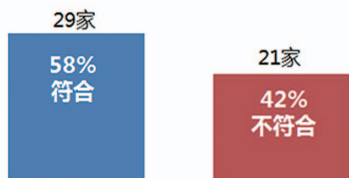
13) 登录中的恶意滥用问题尚无非常有效的解决办法

恶意滥用攻击多发生在 USBKEY 不参与登录的情况下，攻击者针对获得的银行卡号或登录昵称刻意多次输入错误密码，导致被攻击的用户账号锁定。少数银行针对此类锁定需要用户本人携带有

D.3.1

近半数网上银行HTTPS应用实现方式不符合合规要求

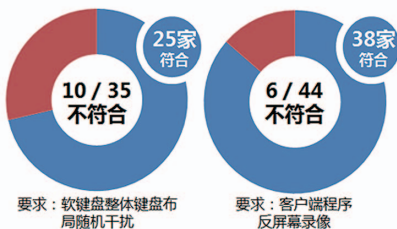
要求：应使用强壮的加密算法和安全协议



备注：1) 数据样本为中国50大银行。2) 百分数采取四舍五入取整。
来源：个人网上银行登录安全研究（2013年1月），绿盟科技

D.3.2

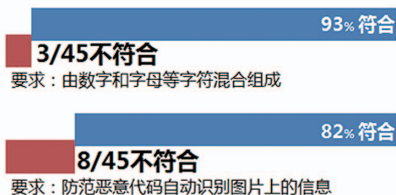
大多数网上银行软键盘或安全控件符合合规要求



备注：数据样本为中国50大银行
来源：个人网上银行登录安全研究（2013年1月），绿盟科技

D.3.3

少数网上银行验证码不符合合规要求



备注：1) 数据样本为中国50大银行。2) 百分数采取四舍五入取整。
来源：个人网上银行登录安全研究（2013年1月），绿盟科技

效证件到柜台办理解锁操作。目前对于有针对性的恶意的账号锁定攻击尚无较有效的防护措施。

14) 用户身份假冒是当前登录面临的最头痛问题

目前银行针对身份假冒攻击的防护，主要从登录过程和登录反馈两个方面进行：其一是采用 HTTPS 协议利用数字证书进行身份验证，对“中间人攻击”进行提示并结合安全控件技术对敏感数据进行加密；其二是利用登录成功短信提醒和显示上次登录信息的方式及时提醒用户可能存在的假冒登录。

通过调研与分析认为：采用 HTTPS 结合 USBKEY 数字证书技术，严格遵守 SSL 过程进行双向身份鉴别的防护效果较好，基本杜绝中间人攻击。

监管：合规不是终点而是起点，不要输在起跑线

目前较为详细、可参考价值最高的网上银行合规要求是由中国人民银行于 2012 年 5 月 8 日发布的《网上银行系统信息安全通用规范》（以下简称《通用规范》）。依此规范，将登录相关的主要安全内容从三方面展开讨论，即网络通信、安全控件和软键盘以及图形验证码。

15) 加密通信协议已经普及，弱加密算法尚存

对于数据传输过程来说，最重要的就是数据信道的安全。防窃听、防劫持刻不容缓。要解决该问题，就取决于传输协议的安全性以及加密算法的强壮程度。

在对网上银行支持的加密算法进行合规分析后发现，有多家银行不同程度地提供了对 SSL 弱加密算法的支持，将使数据传输的安全性大大降低（见图 D.3.1）。

16) 控件和软键盘广泛使用，但技术细粒度有待加强

我们的调研结果显示：8% 的银行未采取任何有效措施来保护用户的输入信息，一旦用户的终端遭受网上银行盗号程序的攻击，用户账号、密码等隐私将直接暴露给攻击者。

针对网上银行控件及软键盘的安全防护方式，《通用规范》中提出了详细的要求。

在软键盘防护技术方面，部分银行未按照《通用规范》要求对键盘布局进行随机干扰或排序，有些银行只对数字键盘部分进行了随机排序但随机性不强。多数单纯使用软键盘方式的银行未能防范屏幕录像技术，因此也为攻击者直接获取到用户账号、密码带来了可能。（见图 D.3.2）

17) 图形验证码安全合规性不容乐观

图形验证码是最好的识别网上银行用户正常登录行为与大规模恶意自动化登录脚本攻击的方式。《通用规范》中，对图形验证码在复杂程度、随机度、图片干扰、使用时间和生成方式方面提出了详细具体的要求。

在图形验证码设计方面，验证码内容构成和抗 OCR 能力都有少数银行不符合要求。在验证码功能实现方面，也存在不同程度的问题，如验证码更新机制、失效机制、验证码有效期等。

局限性：技术也有不足，得与失都得自己承担

在本报告中提到的“九类”个人网上银行安全防护措施中，根据技术与实现的普遍性，选取身份鉴别、输入保护、验证码这三类主要技术应用进行分析和探讨。

18) 利用 USBKEY 证书的登录身份鉴别，也有硬伤

当前网上银行的 USBKEY 硬件主要针对证书安全的存放、数据的加密、身份的认证等方面进行应用，大大提高了网上银行使用的安全性。

USBKEY 存在缺点如下：

1. 用户使用后没有立即拔出 USBKEY，那么攻击者完全可能在这个间歇期伪造一次交易
2. 设计人员在 COS 上留了后门，KEY 内部的密钥不再安全。
3. 当第三方 CA 机构发生安全事故时，证书将变得不可信。
4. 数据从计算机传入 USBKEY 的过程存在被拦截和修改风险。

19) 安全控件是攻与防博弈的矛盾体

当前多数安全控件仅防止了键盘消息钩子，但对通过 IE 的 COM 接口获取密码的方法也无能为力。也有一部分安全控件做得不够底层，技术细节上不够深入，功能还不够完善。

20) 验证码让用户慢下来，影响输入体验

当前 90% 的个人网上银行登录采用了验证码，然而由于抗 OCR 能力的需要，各种复杂的验证码有时难于辨认，很大程度上影响了用户输入体验，拖长了用户登录时间。

当前验证码的实现机制还需要进一步的深化研究，特别是一些缺乏安全思考的开发者错误的实现验证码，将生成图片的信息直接存放于用户的 Cookies、URL 中，更有甚者将验证码算法写于 Web 网页中，这些都是不安全、不正确、风险极大的实现方式。

下一代边界安全专家绿盟科技下一代防火墙正式发布



产品效果图

2013年4月25日是绿盟科技成立十三周年的日子，也是绿盟科技下一代防火墙正式发布的日子。该产品结合绿盟科技十三年攻防研究、产品研发与应用、客户服务经验，历时多载，汇聚近百位专业人士的智慧，精心打造完成。绿盟科技下一代防火墙产品以掌控应用风险、重塑边界安全为目标，在企业网络边界建立VIP式安全防护，即可视化应用安全（Visualization）、一体化安全防护（Integration）、高安全处理性能（Performance），通过智能化识别、精细化控制、一体化扫描等逐层递进的方式实现用户/应用行为可视、可控、合规和安全，最终保障网络应用被安全高效的使用。

随着 Web 2.0 时代的到来，大量网络应用不断涌现，这些应用提高了生产效率，

但同时也带来了许多新的安全威胁，比如恶意软件入侵、网络带宽消耗、机密资料外泄等。而另一方面，面对新的安全威胁，无论是传统防火墙还是统一威胁管理设备（UTM），均已远远不能满足用户对自身网络的安全防护诉求。在此现状下，Gartner 提出了下一代防火墙的定义：下一代防火墙须在兼容传统防火墙的所有功能的同时，结合入侵防护功能、全面精准的用户识别手段、专业的应用识别能力和高效的业务处理能力。

绿盟科技下一代防火墙在满足 Gartner 定义的同时，又融入了多种业界领先技术——业务处理双引擎、一体化安全策略、多核并行处理。其中，业务处理双引擎实现了基础防护、应用层防护分离，保障了业务的永续，结合多核并行处理技术大大提升了网络层性能和应用处理性能；独有的一体化安全策略配置简便、易操作，令使用者工作效率大幅提升。

绿盟科技迄今为止已成功发布多款安全精品，如抗拒绝服务系统（ADS），在全国总共部署容量超过 2000G，且远销日本、美国，为国内抗拒绝服务产品进军海外市场树立了

良好的榜样；又如绿盟科技入侵防护系统（IPS），该产品连续多年位居入侵防护硬件市场第一名，于 2010 年顺利通过了代表全球信息安全领域权威测评机构之一的 NSS Labs 的测试，并获得 Recommended 的最高级别认证；绿盟科技其他产品如 Web 应用防火墙（WAF）、远程安全评估系统（RSAS）等均在国内占据领军的位置。

结合诸多的产品优势和丰富的产品功能，绿盟科技下一代防火墙为用户带来了极大的客户利益，将为客户在下一代网络及需求发展中提供更为全面的安全保障。

重新构建安全——绿盟科技举办第五届信息安全高级论坛

近日，“2013 信息安全高级论坛 —— RSA conference 2013 热点研讨”在北京



召开。本届论坛由中关村科技园区管理委员会、中国计算机学会计算机安全专业委员会主办，绿盟科技承办，旨在搭建一个国际信息安全技术、趋势的交流平台，分享国际信息安全热点。与此同时，本届论坛还结合国内信息安全行业的现状，融入本土化的思考，在去年“重新思考安全”的基础上，与会嘉宾以“重新构建安全”为主题进行了探讨。中国工程院沈昌祥院士以及中关村科技园区管理委员会、公安部、工业和信息化部、中国信息安全测评中心的有关领导出席了本次大会并致辞。

信息安全行业正处于错综变化的发展时期，无论是从各种国际会议中呈现的高级持续威胁（APT）、网络战、网际安全、云安全等热点话题，还是从国内下一代安全产品的热产热销、云计算的蓬勃发展，都能感受到行业正处在变革期。在今年的论坛上，业内专家学者以“重新构建安全”为主题，从理论思考、市场变化以及新技术应用等方面发表了他们的看法，并分享了最新研究成果。

中国科学院信息工程研究所 DCS 中心

翟起滨院士以“如何面对我们在网际空间的实际位置”为主题，对现阶段我国信息网络安全处境与现状进行了探讨。清华大学信息技术研究院李军院长从 SDN 的基本思想、学术起源、产业影响、安全问题、发展走向五个方面为到场来宾介绍了 SDN 的演进与网络安全变革。北京大学信息学院计算机系陈钟主任从学科建设与人才培养的方向，谈到了网络与信息安全的人才培养模式。国家网络信息安全技术研究所杜跃进所长就新型网络安全能力建设的需求提出了若干技术思路。

作为信息安全产业的企业代表，华为、绿盟科技也从各自关注的角度在产业变化、市场变化与技术发展的层面分享了各自的观点。华为郑志彬博士与来宾分享了从 RSA 看信息安全业务变化发展趋势。绿盟科技副总裁吴云坤从下一代安全的视角解读了信息安全最新发展趋势以及绿盟科技的应对策略。

信息安全行业正面临一场新形势、新变革，绿盟科技始终站在技术探索的最前沿，不断摸索信息安全的前沿科技，并致力于信

息安全技术的推广与应用。

绿盟科技荣获 Web 应用综合防护杰出贡献奖

在日前举办的“第五届通信网络和信息安全高层论坛”上，绿盟科技荣获 Web 应用综合防护杰出贡献奖。本次论坛吸引了众多政府主管、运营商代表、安全企业专家，会上就网络信息安全的新趋势、移动互联网终端安全趋势与应对、Web 安全、云服务等议题进行了交流。

在本次大会上，绿盟科技产品经理李从宇引述了绿盟科技研究院的两份报告《Operation Ababil 事件分析》和《2012 绿盟科技威胁态势报告》，阐述了数据中心轮番遭遇 DDoS 和 Web 攻击的微观发生过程和宏观发展趋势，提出了数据中心防御 DDoS 和 Web 攻击的新架构。新架构紧紧盯住攻击者，做到了同时抑制 DDoS 和 Web 攻击，还可以随“击”应变，节省投资。随着数据中心安全服务向精细化方向发展，在攻击者轮番使用 DDoS 和 Web 攻击时，数据中心如何快速应对，正得到业内的广泛关注。

NSFOCUS 2013年2月之十大安全漏洞

声明：本十大安全漏洞由 NSFOCUS(绿盟科技) 安全小组 <security@nsfocus.com> 根据安全漏洞的严重程度、利用难易程度、影响范围等因素综合评出，仅供参考。
http://www.nsfocus.net/index.php?act=sec_bug&do=top_ten

1. 2013-02-19 Adobe Flash Player 缓冲区溢出漏洞 (CVE-2013-0633)(APSB13-04)

NSFOCUS ID: 22622

<http://www.nsfocus.net/vulndb/22622>

综述：

Adobe Flash Player 是一个集成的多媒体播放器。

Adobe Flash Player 没有正确处理特制的 SWF 内容，在实现上存在缓冲区溢出漏洞，可允许远程攻击者执行任意代码。

危害：

攻击者可以通过诱使受害者打开恶意 SWF 来利用此漏洞，从而控制用户系统。

2. 2013-02-19 Adobe Flash Player 远程内存破坏漏洞 (CVE-2013-0634)(APSB13-04)

NSFOCUS ID: 22640

<http://www.nsfocus.net/vulndb/22640>

综述：

Adobe Flash Player 是一个集成的多媒体播放器。

Adobe Flash Player 没有正确处理特制的 SWF 内容，可允许远程攻击者执行任意代码或造成内存破坏，拒绝服务。

危害：

攻击者可以通过诱使受害者打开恶意 SWF 来利用此漏洞，从而控制用户系统。

▶ 安全公告

3. 2013-02-27 Adobe Flash Player 远程代码执行漏洞 (CVE-2013-0648)

NSFOCUS ID: 22748

<http://www.nsfocus.net/vulndb/22748>**综述：**

Adobe Flash Player 是一个集成的多媒体播放器。

Adobe Flash Player 的 ExternalInterfaceActionScript 功能在实现上存在安全漏洞。

危害：

攻击者可以通过诱使受害者打开恶意 SWF 来利用此漏洞，从而控制用户系统。

4. 2013-02-27 Adobe Flash Player 不明细节安全漏洞 (CVE-2013-0643)

NSFOCUS ID: 22747

<http://www.nsfocus.net/vulndb/22747>**综述：**

Adobe Flash Player 是一个集成的多媒体播放器。

Adobe Flash Player 内的 Firefox 沙盒没有正确限制权限，可使远程攻击者通过特制的 SWF 内容执行任意代码。

危害：

攻击者可以通过诱使受害者打开恶意 SWF 来利用此漏洞，从而控制用户系统。

5. 2013-02-19 Adobe Acrobat 和 Reader 远程代码执行漏洞 (CVE-2013-0640)(APSA13-02)

NSFOCUS ID: 22702

<http://www.nsfocus.net/vulndb/22702>**综述：**

Adobe Reader 是美国 Adobe 公司开发的一款优秀的 PDF 文档阅读软件。

Adobe Reader/Acrobat 在实现上存在漏洞，允许远程攻击者通过特制的 PDF 文档执行任意代码。

危害：

攻击者可以通过诱使受害者打开恶意 PDF 来利用此漏洞，从而控制用户系统。

6. 2013-02-19 Adobe Acrobat 和 Reader 远程代码执行漏洞 (CVE-2013-0641)(APSA13-02)

NSFOCUS ID: 22621

<http://www.nsfocus.net/vulndb/22621>**综述：**

Adobe Reader 是美国 Adobe 公司开发的一款优秀的 PDF 文档阅读软件。

Adobe Reader/Acrobat 在实现上存在漏洞，允许远程攻击者通过特制的 PDF 文档绕过沙盒限制。

危害：

攻击者可以通过诱使受害者打开恶意 PDF 来利用此漏洞，从而控制用户系统。

7. 2013-02-18 Microsoft Internet Explorer Vector Markup Language 内存破坏漏洞 (CVE-2013-0030) (MS13-010)

NSFOCUS ID: 22612

<http://www.nsfocus.net/vulndb/22612>**综述：**

Microsoft Internet Explorer 是微软公司推出的一款网页浏览器。

Microsoft Internet Explorer 6 - 10 版本内的 VML 实现中没有正确分配缓冲区，存在漏洞。

危害：

攻击者可以通过诱使受害者打开恶意网页来利用此漏洞，从而控制用户系统。

8. 2013-02-26 Oracle Java Runtime Environment 多个不明细节远程代码执行漏洞

NSFOCUS ID: 22741

<http://www.nsfocus.net/vulndb/22741>**综述：**

Oracle Java Runtime Environment (JRE) 是一款为 Java 应用程序提供可靠运行环境的解决方案。

Oracle Java Runtime Environment (JRE) 在实现上存在多个不明细节远程代码执行漏洞，攻击者可利用这些漏洞在应用上下文中执行任意代码。

危害：

攻击者可以通过诱使受害者打开恶意网页来利用此漏洞，从而控制用户系统。

9. 2013-02-19 ECShop 支付宝插件 SQL 注入漏洞

NSFOCUS ID: 22626

<http://www.nsfocus.net/vulndb/22626>**综述：**

ECSHOP 是开源的网店系统。

ECSHOP 支付插件存在高危 SQL 注入漏洞，这个漏洞是一个 0day 漏洞。

危害：

攻击者可利用 SQL 注入绕过系统限制获取网站数据，进而实施“拖库”窃取网站资料。

10. 2013-02-26 Linux kernel sock_diag 越界访问 sock_diag_handlers[] 本地权限提升漏洞

NSFOCUS ID: 22727

<http://www.nsfocus.net/vulndb/22727>**综述：**

Linux Kernel 是 Linux 操作系统的内核。

Linux kernel 3.3 - 3.8 及其他版本允许未授权用户发送 netlink 消息，越界访问 sock_diag_handlers[] 数组，导致 userland 在内核模式中得到控制权。

危害：

本地攻击者可以利用这个漏洞进行权限提升。

NSFOCUS 2013年3月之十大安全漏洞

声明：本十大安全漏洞由 NSFOCUS(绿盟科技) 安全小组 <security@nsfocus.com> 根据安全漏洞的严重程度、利用难易程度、影响范围等因素综合评出，仅供参考。
http://www.nsfocus.net/index.php?act=sec_bug&do=top_ten

1. 2013-03-20 Microsoft Internet Explorer 沙盒保护机制内存破坏漏洞 (CVE-2013-2557)

NSFOCUS ID: 23033

<http://www.nsfocus.net/vulndb/23033>

综述：

Microsoft Internet Explorer 是微软公司推出的一款网页浏览器。

Microsoft Internet Explorer 9 的沙盒保护机制存在未知细节安全漏洞，可允许远程攻击者造成拒绝服务（内存破坏）或其他影响。

危害：

攻击者可以通过诱使受害者打开恶意网页来利用此漏洞，从而控制受害者系统。

2. 2013-03-20 Microsoft Windows ASLR 安全绕过漏洞

(CVE-2013-2556)

NSFOCUS ID: 23032

<http://www.nsfocus.net/vulndb/23032>

综述：

Microsoft Windows 是微软公司推出的一系列操作系统。

Microsoft Windows 7 内存在不明细节漏洞，可允许攻击者绕过 ASLR 保护机制。

危害：

攻击者可以利用此漏洞绕过系统的 ASLR 保护机制，提升攻击的成功率。

3. 2013-03-13 Adobe Flash Player/AIR 释放后重用远程代码执行漏洞 (CVE-2013-0650)

NSFOCUS ID: 22939

<http://www.nsfocus.net/vulndb/22939>

综述：

Adobe Flash Player 是一个集成的多媒体播放器。

Adobe Flash Player 及 AIR 在实现上存在释放后重用漏洞，该漏洞可导致代码执行。

危害：

攻击者可以通过诱使受害者打开恶意 SWF 文件来利用此漏洞，从而控制受害者系统。

4. 2013-03-20 Adobe Reader 远程代码执行漏洞 (CVE-2013-2550)

NSFOCUS ID: 23054

<http://www.nsfocus.net/vulndb/23054>

综述：

Adobe Reader(也被称为 Acrobat Reader) 是美国 Adobe 公司开发的一款优秀的 PDF 文档阅读软件。

Adobe Reader 11.0.02 及其他版本在实现上存在不明细节漏洞，可允许攻击者绕过沙盒保护机制。

危害：

攻击者可以通过诱使受害者打开恶意 PDF 文件来利用此漏洞，绕过沙盒系统。

5. 2013-03-11 Mozilla Firefox/Thunderbird/SeaMonkey 远程代码执行漏洞 (CVE-2013-0787)

NSFOCUS ID: 22909

<http://www.nsfocus.net/vulndb/22909>

综述：

Firefox 是一款非常流行的开源 Web 浏览器。SeaMonkey 是开源的 Web 浏览器、邮件和新闻组客户端、IRC 会话客户端和 HTML 编辑器。Thunderbird 是一个邮件客户端，支持 IMAP、POP 邮件协议以及 HTML 邮件格式。

Mozilla Firefox、SeaMonkey、Thunderbird 在内部编辑器操作进行时，函数 document.execCommand() 运行内容脚本过程中，HTML 编辑器内存在释放后重用漏洞，可导致任意代码执行。

危害：

攻击者可以通过诱使受害者打开恶意网页来利用此漏洞，从而控制受害者系统。

6. 2013-03-11 WebKit Type Confusion 远程代码执行漏洞 (CVE-2013-0912)

NSFOCUS ID: 22908

<http://www.nsfocus.net/vulndb/22908>

综述：

WebKit 是一个开源的浏览器引擎。

Google Chrome 25.0.1364.160 之前版本在 WebKit 的实现上存在类型混淆错误。

危害：

▶▶ 安全公告

攻击者可以通过诱使受害者打开恶意网页来利用此漏洞，绕过沙盒系统。

7. 2013-03-04 Oracle JRE 2D Component 处理远程代码执行漏洞 (CVE-2013-1493)

NSFOCUS ID: 22803

<http://www.nsfocus.net/vulndb/22803>

综述：

Oracle Java Runtime Environment (JRE) 是一款为 Java 应用程序提供可靠运行环境的解决方案。

Oracle JRE 1.6.0 Update 41、1.7.0 Update 15 在处理 2D Component 时存在内存破坏漏洞。

危害：

攻击者可以通过诱使受害者打开恶意图像文件来利用此漏洞，从而控制受害者系统。

8. 2013-03-18 Apple Mac OS X 释放后重用远程代码执行漏洞 (CVE-2013-0971)

NSFOCUS ID: 23000

<http://www.nsfocus.net/vulndb/23000>

综述：

Apple Mac OS X 是苹果电脑操作系统软件。

Apple Mac OS X 10.8.3 之前版本 PDFKit 内存存在释放后重用漏洞，通过 PDF 文档内的特制链接注释，远程攻击者可执行任意代码或造成拒绝服务。

危害：

攻击者可以通过诱使受害者打开恶意文件来利用此漏洞，绕过沙盒系统。

9. 2013-03-14 Linux Kernel 'CLONE_NEWUSER|CLONE_FS' 本地权限提升漏洞

NSFOCUS ID: 22968

<http://www.nsfocus.net/vulndb/22968>

综述：

Linux Kernel 是 Linux 操作系统的内核。

Linux kernel 在 CLONE_NEWUSER|CLONE_FS 的实现上存在本地权限提升漏洞。

危害：

本地攻击者可利用此漏洞获取内核权限。

10. 2013-03-20 EA Origin 游戏服务平台任意代码执行漏洞

NSFOCUS ID: 23039

<http://www.nsfocus.net/vulndb/23039>

综述：

EA Origin 是一个在线的网络游戏平台。

EA Origin 在 URI 处理机制上存在安全漏洞，可允许攻击者在玩家的电脑上远程执行恶意代码。

危害：

远程攻击者可以利用此漏洞控制受害者系统。

NSFOCUS 2013年4月之十大安全漏洞

声明 : 本十大安全漏洞由 NSFOCUS(绿盟科技) 安全小组 <security@nsfocus.com> 根据安全漏洞的严重程度、利用难易程度、影响范围等因素综合评出, 仅供参考。
http://www.nsfocus.net/index.php?act=sec_bug&do=top_ten

1. 2013-04-10 Microsoft Internet Explorer 释放后重用远程代码执行漏洞 (CVE-2013-1303) (MS13-028)

NSFOCUS ID: 23244

<http://www.nsfocus.net/vulndb/23244>

综述 :

Microsoft Internet Explorer 是微软公司推出的一款网页浏览器。

Microsoft Internet Explorer 6/7/8/9/10 访问内存中已被删除的对象时存在远程执行代码漏洞。

危害 :

攻击者可以通过诱使受害者打开恶意网页来利用此漏洞, 从而控制用户系统。

2. 2013-04-10 Adobe Flash Player 和 AIR 内存破坏漏洞 (CVE-2013-1378)

NSFOCUS ID: 23279

<http://www.nsfocus.net/vulndb/23279>

综述 :

Adobe Flash Player 是一个集成的多媒体播放器。

Adobe Flash Player 存在整数溢出安全漏洞, 此漏洞可导致远程代码执行。

危害 :

攻击者可以通过诱使受害者打开恶意 SWF 文件来利用此漏洞, 从而控制用户系统。

3. 2013-04-23 Oracle Java Runtime Environment 'Reflection API' 远程代码执行漏洞

NSFOCUS ID: 23561

<http://www.nsfocus.net/vulndb/23561>

▶▶ 安全公告

综述：

Oracle Java Runtime Environment(JRE) 是运行 Java 程序所必须的环境的集合。

Oracle Java Runtime Environment 1.7.0_21-b11 及之前版本、JDK 软件存在安全漏洞，可导致绕过目标系统上的 Java 安全沙盒。

危害：

攻击者可以通过诱使受害者打开恶意网页来利用此漏洞，从而控制用户系统。

4. 2013-04-16 Google Chrome OS 26.0.1410.57 之前版本多个安全漏洞

NSFOCUS ID: 23343

<http://www.nsfocus.net/vulndb/23343>

综述：

Google Chrome OS 是一款基于 Linux 的开源操作系统。

Google Chrome OS 26.0.1410.57 之前版本存在多个安全漏洞，包括未初始化的内存数据、释放后重用锁定绕过等。

危害：

攻击者可以利用这些漏洞控制受害者系统。

5. 2013-04-03 Mozilla Firefox/Thunderbird/SeaMonkey 多个漏洞 (MFSA 2013-30-40)

NSFOCUS ID: 23174

<http://www.nsfocus.net/vulndb/23174>

综述：

Firefox 是一款非常流行的开源 Web 浏览器。SeaMonkey 是开源的 Web 浏览器、邮件和新闻组客户端、IRC 会话客户端和 HTML 编辑器。Thunderbird 是一个邮件客户端。

Mozilla Firefox、Thunderbird、SeaMonkey 在实现上存在多个安全漏洞，可以导致应用崩溃、获取敏感信息、提升权限、绕过安全限制、执行未授权操作等。

危害：

攻击者可以通过诱使受害者打开恶意网页来利用此漏洞，从而控制受害者系统。

6. 2013-04-25 WebKit “CompositeEditCommand.cpp” 释放后重用远程代码执行漏洞

NSFOCUS ID: 23557

<http://www.nsfocus.net/vulndb/23557>

综述：

WebKit 是一个开源的浏览器引擎，也是苹果 Mac OS X 系统引擎框架版本的名称。

WebKit 在克隆段时，WebCore/editing/CompositeEditCommand.cpp 里的函数“CompositeEditCommand::cloneParagraphUnderNewElement”存在释放后重用错误。

危害：

攻击者可以通过诱使受害者打开恶意网页来利用此漏洞，从而控制受害者系统。

7. 2013-04-08 Cogent Real-Time Systems 多个产品栈缓冲区溢出漏洞 (CVE-2013-0680)

NSFOCUS ID: 23218

<http://www.nsfocus.net/vulndb/23218>

综述：

Cogent Real-Time Systems 是加拿大厂商，生产与控制系统交互的中间件应用。

Cogent Real-Time Systems Cogent DataHub、OPC DataHub 等产品存在栈缓冲器溢出安全漏洞。

危害：

远程攻击者通过发送较长的 HTTP 标头来利用此漏洞，导致远程服务器拒绝服务或执行任意代码。

8. 2013-04-26 phpMyAdmin preg_replace() 远程 PHP 代码执行 (CVE-2013-3238)

NSFOCUS ID: 23576

<http://www.nsfocus.net/vulndb/23576>

综述：

phpMyAdmin 是 MySQL 数据库的在线管理工具。

phpMyAdmin 3.5.8、4.0.0-rc2 及其他版本的 preg_replace() 函数可被利用在服务器端执行任意 PHP 代码。

危害：

攻击者可以通过向服务器提交恶意请求来利用此漏洞，从而控制服务器系统。

9. 2013-04-11 Cisco IOS XE IPv6 Multicast 和 IPv6 MVPN Traffic 远程拒绝服务漏洞

NSFOCUS ID: 23301

<http://www.nsfocus.net/vulndb/23301>

综述：

Cisco IOS 是多数思科系统路由器和网络交换机上使用的互联网络操作系统。

Cisco 1000 Series ASR 在处理碎片 IPv6 多播流量和破碎的 IPv6 MVPN 报文时存在漏洞。

危害：

会让未经身份验证的远程攻击者造成设备重载，导致拒绝服务。

10. 2013-04-10 Microsoft Windows OpenType 字体解析远程拒绝服务漏洞 (CVE-2013-1291)(MS13-036)

NSFOCUS ID: 23249

<http://www.nsfocus.net/vulndb/23249>

综述：

Microsoft Windows 是微软公司推出的一系列操作系统。

Windows 未能处理特制字体文件，存在一个拒绝服务漏洞。此漏洞可能会导致计算机停止响应和重新启动。

危害：

攻击者可以通过诱使受害者打开畸形字体文件来利用此漏洞，从而控制受害者系统。

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。

“随着企事业信息化程度提高，如何管理IT风险、使IT更好地为企事业战略服务显得至关重要。”

孙晓鹏

绿盟科技北京分公司 安全顾问



★ 为了更加及时的应对危机，绿盟科技的服务与销售网络现已遍布全国；无论何时何地，绿盟科技的安全专家都能为您提供同样卓越的安全解决方案与服务。



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

www.nsfocus.com



THE EXPERT BEHIND GIANTS 巨人背后的专家