网络安全

安全技巧（九）

小未 主编

目　录

浅析网络入侵监测系统-IDS 的应用

很多文章介绍了如何通过建立，改善，以及分析服务器日记文件的种种方式，监测出来黑客入侵行为，但这些都是过去式，都是在入侵发生后你才知道存在这种行为而加以防范。最好的方法是能够在当场就能监测出恶意的网络入侵行为，并且马上采取防范反击措施加以纠正。因此即时监测黑客入侵行为并以程序自动产生响应的网络入侵监测系统（又称 IDS）产生了。

1、何谓 IDS？

简单的说，设立 IDS 的唯一目的就是当场监测到网络入侵事件的发生。IDS 就是一个网络上的系统，这个系统包含了下面三个组件：

（1）网络监测组件，用以捕捉在网络线上传递的封包。

（2）接口组件，用以决定监测中的资料传递是否属于恶意行为或恶意的使用。在网络传递时，用来比较的资料样式 （pattern），以监测恶意网络活动。

（3）响应组件，针对当时的事件予以适当的响应。这个响应可以是简单的，例如寄发一个电子邮件讯息给系统管理者，或者是复杂的，例如暂时将违规者的 IP 地址过滤掉，不要让他连到这个网络来。

2、IDS 如何通过网页监测网络入侵事件

IDS 系统不只必须监测各式各样，从大到小，以及各种系列的系统上的网络攻击事件，它还必须能够快速及时地的在第一时间内监测到入侵事件的发生。因此，I

DS 的数据库以及式样比对（pattern-matching）机制是复杂到令人难以置信的。

要使 IDS 能够监测通过网页的入侵事件，其中的网络监测组件就必须要能够捕捉所有通过网页通讯端口上，借着 HTTP 通讯协议传递的网络资料往来。（注意，SSL 的网络交通是完全绕过 IDS 的网络监测的，因为这些网络交换资料都是经过加密的。）式样比对组件在这里，主要是用于比较 URL 解析的结果，看看是否符合数据库中的恶意的 HTTP 回询（request）。

接下来，我介绍如何制作两个快速而简易的 IDS，用来监测可疑的网页回询活动。这些解决方案的目的是在于提供系统管理者，让他们拥有一个特别针对他们网络而设计的监测/响应系统。

3、制作快速而简易的 IDS

（1）Network Grep 工具

我们先从一个简单的网络监视程序开始，这个程序是用来监测 HTTP 通讯协议的网络资料往来。HTTP 回询的特色是，它使用以下的语法：

〈HTTP-Request-Method〉 〈URL〉 HTTP/〈version〉

这个可在 Packetfactory 入口网站寻获的程序 ngrep 针对在网络上传递往来的资料，执行正则表示法（regular expression）式样比对。我们可以用以下的指令来利用 ngrep 拦截并显示所有纯文字形式的 HTTP 资料往来：

#ngrep-iqt“^GET|^HEAD|^TRACE|^POST|^PUT and HTTP”

以上指令中，-iqt 选项是指示 ngrep 不要区分资料

中的大小写，并且只有显示封包中有符合式样比对的资料，以及在显示资料时加上日期以及时间的标题。（注：比对的式样，是基于 GET，HEAD，TRACE，POST，PUT，以及 HTTP 等关键词。欲知更多有关如何在 ngrep 使用正则表示法，你可以到 http://www.packetfactory.net/Projects/Ngrep/查看相关资料。）

以上面我们建议的方式使用 ngrep 再加上运行越来越受欢迎的 Whisker 程序，监测地址为 10.1.1.2 的 IIS 5.0 服务器平台，我们得到了以下的结果：

T 03:37:30.041739 10.1.1.21:2425 -> 10.1.1.2:80 [AP]

HEAD / HTTP/1.0..User-Agent: Mozilla/5.0 [en]（Win95; U）..Referer: http://10.1.1.2/..Connection: close....

T 2001/01/16 03:37:30.108630 10.1.1.21:2426 -> 10.1.1.2:80 [AP]

GET /cfdocs/ HTTP/1.0..User-Agent: Mozilla/5.0 [en]（Win95; U）..Cookie: ASPSESSIONIDGQGQGLAC=HDJNBOGBIPOCPNCKOJOPBCFD;path=

/..Referer:http://10.1.1.2/..Connection: close....

T 2001/01/16 03:37:31.842452 10.1.1.21:2427 -> 10.1.1.2:80 [AP]

GET /scripts/ HTTP/1.0..User-Agent: Mozilla/5.0 [en]（Win95; U）..Cookie: ASPSESSIONIDGQGQGLAC=HDJNBOGBIPOCPNCKOJOPBCFD;path=

/..Referer:http://10.1.1.2/..Connection: close....

T 2001/01/16 03:37:31.854206 10.1.1.21:2428 -> 10.1.1.2:80 [AP]

GET /scripts/cfcache.map HTTP/1.0..User-Agent: Mozilla/5.0 [en]

（Win95; U）..Cookie: ASPSESSIONIDGQGQGLAC=HDJNBOGBIPOCPNCKOJOPBCFD;

path=/..Referer: http://10.1.1.2/..Connection: close....

T 2001/01/16 03:37:33.644534 10.1.1.21:2429 -> 10.1.1.2:80 [AP]

GET /cfcache.map HTTP/1.0..User-Agent: Mozilla/5.0 [en] （Win95; U）..Cookie: ASPSESSIONIDGQGQGLAC=HDJNBOGBIPOCPNCKOJOPBCFD;path=

/..Referer:http://10.1.1.2/..Connection: close....

现在你就可以采取行动了！

（2）执行式样比对

使用 ngrep **拦截网络资料往来很简单。然而，分析捕捉到的资料并从中抽取** URL **则略具难度。因为** ngrep **将资料输出拆成一行一行的，所以我们必须额外耗费很多精力，去重组输出的资料，并将该资料中的** URL **与已知的网络攻击行为模式做比对。**

此时，我向大家介绍另一个用来监测网页传送的犀利工具软件了。这个软件就叫做 urlsnarf**，它是由** Dug Song **写成的** dsniff **工具软件套件的一部份。**urlsnarf **从所拦截的网络资料传送中，捕捉所有的** HTTP **回询，并且将结果以共享日记文件格式（**Common Log Format , CLF**）显示出来，这种格式就跟市面上的网页服务器，诸如** Apache **或者是** IIS **所用的格式一样。**

跟当初我们用 ngrep **的方式一样，我们使用** urlsnarf **并且在** 10.1.1.2 **的服务器上执行** Whisker**，所得到的**

结果如下：

urlsnarf

urlsnarf: listening on eth0

10.1.1.21 - - [16/02/2001:03:58:43 +0530] "HEAD http://10.1.1.2/ HTTP/1.0" - - "http://10.1.1.2/" "Mozilla/5.0 [en] （Win95; U）"

10.1.1.21 - - [16/02/2001:03:58:43 +0530] "GET http://10.1.1.2/cfdocs/ HTTP/1.0" - - "http://10.1.1.2/" "Mozilla/5.0 [en] （Win95; U）"

10.1.1.21 - - [16/02/2001:03:58:45 +0530] "GET http://10.1.1.2/scripts/ HTTP/1.0" - - "http://10.1.1.2/" "Mozilla/5.0 [en] （Win95; U）"

10.1.1.21 - - [16/02/2001:03:58:45 +0530] "GET http://10.1.1.2/scripts/cfcache.map HTTP/1.0" - - "http://10.1.1.2/" "Mozilla/5.0 [en] （Win95; U）"

10.1.1.21 - - [16/02/2001:03:58:48 +0530] "GET http://10.1.1.2/cfcache.map HTTP/1.0" - - "http://10.1.1.2/" "Mozilla/5.0 [en] （Win95; U）"

10.1.1.21 - - [16/02/2001:03:58:50+0530]"GET http://10.1.1.2/cfide/Administrator/startstop.html HTTP/1.0" - - "http://10.1.1.2/" "Mozilla/5.0 [en] （Win95; U）"

10.1.1.21 - - [16/02/2001:03:58:52 +0530] "GET http://10.1.1.2/cfappman/index.cfm HTTP/1.0" - - "http://10.1.1.2/" "Mozilla/5.0 [en] （Win95; U）"

使用 urlsnarf **唯一的缺点是，它现在的程序是写死的，只监听** TCP **通讯端口** 80（**纯文字** HTTP），3128（MS-proxy）**以及** 8080（generic/squid proxy）。**从其它通讯**

端口传输的 HTTP 协议资料则完全被忽略。要想改变这种限制，你必须在 urlsnarf 的原始程序代码中做一些小小的改变。然而，光是 urlsnarf 所提供的功能，就已经远远的超过它所给我们的限制了。

因为 urlsnarf 以 CLF 格式产生日记，我们可以将它的输出结果，转送到任何在网页服务器上使用 CLF 格式分析日记的日记分析软件。

4、监测恶性入侵性网页浏览行为

通过 urlsnarf 的输出，我们可以开始建立式样比对程序，以寻找网络入侵事件。在这里我利用一个简单的 Perl 程序来跟 urlsnarf 一起监测一些基本的网络入侵行为。我们会把 urlsnarf 的执行结果转传给这个式样比对程序，通过式样比对的方法监测网络入侵行为。

式样比对程序的第一步是，定义一连串入侵性的 URL 查询。为了简单起见，我们只列出某些 URL 如下:

```
%cgis = (
"/msadc/msadcs.dll" => "mdac",
"/msadc/Samples/selector/showcode.asp" => "showcode",
"/cgi-bin/guestbook.cgi" => "guestbook",
"/cgi-bin/test-cgi" => "test-cgi",
"/cgi-bin/finger" => "finger",
"/cfdocs/expelval/exprcalc.cfm" => "exprcalc",
"/cgi-bin/phf" => "phf",
"/scripts/samples/search/webhits.exe" => "webhits",
"/scripts/iisadmin/ism.dll" => "ism",
"/scripts/tools/newdsn.exe" => "newdsn",
"/scripts/perl.exe" => "perl_exe",
"/scripts/proxy/w3proxy.dll" => "w3proxy"
```

）；

我们使用了%cg集中储存所有我们需要的恶意URL查询式样。在这里，我们也可以从一个含有这些“特征”的档案，动态建立这个查询式样库。 注意，以上的URL本身并无害；然而，它们通常被黑客利用来做恶意的网页攻击的基础。（例如：msdacs.dll就可以被用来破坏MDAC/RDS）。

下一步，是设定容忍的最低程度，即：如果某个访客查询某个URL超过三次的话，这个访客的IP地址就会被列在黑名单中。在我们的程序里，定义如下：

```
$threshold = 3;
```

下一段重要的程序代码，是一个以while叙述开始的循环，这个循环会从urlsnarf读取每一个CLF纪录，并且做分析。为了避免谈到太多Perl程序语言的细节，有关 while 循环的说明就像以下这样：

```
while（〈 〉） {
 # # parse incoming log line
 # $logline = $_;
 # # pick out the IP，timestamp andURLfrom the CLFline
 # $logline =~ /（S+）.+?（[.+]）.+?（".+?"）.+/;
 # $ip = $1;
 # $time = $2;
 # $url = $3;
 # # select the resource from the URL
 # $url =~ /w+s+.*//.+?（/.*）s+.*/;
 # $resource = $1;
 # check if there is a match with theURL
```

变量$resource 的值为 URL 回询中的 resource 字符串。例如，如果 URL 为 http://10.1.1.2/msadc/msadcs.dll，那么 resource 字符串的值就是 /msadcs/msadcs.dll。

接着是，寻找我们的 URL“特征”库，看看所查询的 URL 字符串是否符合其中的一个特征。如果式样符合，我们找出这个查询出处的 IP 地址， 然后将它的访客指数加一。如果访客观存在指数超过了我们的容忍底线，那么我们将这个 IP 地址标为黑客地址。

下面是式样比对部分的程序代码:

```
# check if there is a match with the URL
if（$cgis{$resource} ne ""） {
push（@{ $offender_list{$ip} }, $cgis{$resource}）;
# check if the threshold count is crossed
if（$offence_count{$ip}++ > $threshold） {
# response to intrusion detected
print STDERR "** $ip " . join（" "，@{ $offend
er_list{$ip} }） . "n";
} }
```

将这个程序取名为 pattern_match.pl。开始使用 urlsnarf 以及 pattern_match.pl，urlsnarf 以及 pattern_match.pl 得出来的结果应该是如下所示:

```
#urlsnarf| pattern_match.pl
```

一个 Whisker 扫描范例，执行 urlsnarf 以及 pattern_match.pl，监测地址为 10.1.1.2 的 IIS5.0 服务器平台，我们得到了以下的结果:

```
** 10.1.1.21 webhits ism showcode newdsn
** 10.1.1.21 webhits ism showcode newdsn mdac
** 10.1.1.21 webhits ism showcode newdsn mdac
```

w3proxy

** 10.1.1.21 webhits ism showcode newdsn mdac w3proxy perl_exe

这些结果告诉我们，来自 IP 地址 10.1.1.21 的访客为恶意访客，并且也列出了一连串针对 10.1.1.2 的相关可疑的 URL 回询。黑客回报系统是在“特征 URL”已经被查询三次了以后，第四次类似的查询又发生（newdsn）才被激活的。

5、小结

在这里，我向大家介绍了如何利用 IDS 系统监测以网页为媒介的网络入侵活动以及示范如何让各式各样的工具以及 script 在很短的时间内组织起来，以形成功能强大的工具。但由于本人的知识及经验有限，难免存在不足之处，希望大家验给予指正，谢谢！在以后在篇章中，我将会大家介绍网络入侵监测软件如何抵挡黑客们使用的 IDS 躲避技术。

多层次的安全防护系统：让黑客选择放弃

随着网络技术的快速发展和应用的日渐普及，黑客工具不仅变得越来越先进，而且也越来越容易被一般人获取和滥用。黑客技术的提升和黑客工具的泛滥，造成大量的企业、机构和个人的电脑系统遭受程度不同的入侵和攻击，或面临随时被攻击的危险。这就迫使大家不得不加强对自身电脑网络系统的安全防护，甚至追求所谓彻底的、一劳永逸的、100%的网络安全解决方案。

但是，网络安全专家和专业厂商强调，没有一个公

司的安全系统能保证 100%的安全。安全总是相对的。本文阐释的所谓“ 多层次防护 ”, 就是应用和实施一个基于多层次安全系统的全面信息安全策略，在各个层次上部署相关的网络安全产品，增加攻击者侵入所花费的时间、成本和所需要的资源，从而卓有成效地降低被攻击的危险，达到安全防护的目标。事实上，多层次防护已经成为当今网络安全的主流策略。

攻击与防范的互动

网络入侵和安全防范实际上就是指网络攻防技术。攻防技术的此消彼涨始终是网络安全领域前进的动力。攻击技术包括目标网络信息收集技术，目标网络权限提升技术，目标网络渗透技术，目标网络摧毁技术四大类。每一类技术，都是日新月异、不断更新的。所以在网络的安全防范上，我们面对越来越多的新技术的攻击。

在用户方面，不管你是否已经受到这些攻击，不管这些攻击是否产生了比较严重的后果，你都必须假设它们对信息系统的威胁总是存在的。因为一旦你的信息系统受到攻击，你就会蒙受无法估量的损失。在任何时候，对信息系统的连续不断的保护是非常必要的。研究分析表明，单一的安全保护往往效果不理想，最佳途径就是采用多层安全防护措施对信息系统进行全方位的保护。

建设安全保护层

“ 分层的安全防护 ”提供了这样一种思路：结合不同的安全保护因素，例如防病毒软件、防火墙和安全漏洞检测工具，来创建一个比单一防护有效得的多的综合的保护屏障。分层的安全防护成倍地增加了黑客攻击的成本和难度，从而大大减少了他们对企业的攻击。

分层的安全防护技术具体来说包括攻击检测，攻击

防范，攻击后的恢复这三个大方向，每一个方向上有代表性的产品：入侵检测系统负责进行攻击检测，防火墙和强制访问控制系统负责攻击防范，攻击后的恢复则由自动恢复系统来解决。这三大方向就说明了在网络安全防护上的多层安全防护措施。

下面我们就多层次保护中包括的主要环节做具体地说明。

入侵检测系统

入侵检测系统是近年出现的新型网络安全技术，目的是提供实时的入侵检测及采取相应的防护手段，如记录证据用于跟踪和恢复、断开网络连接等。

实时入侵检测能力之所以重要，首先因为它能够对付来自内部网络的攻击，其次它能够减少被黑客入侵的时间。基于主机的入侵检测系统用于保护关键应用的服务器，实时监视可疑的连接、系统日志检查，非法访问的闯入等，并且提供对典型应用的监视。

因此，在提供关键服务的服务器上使用入侵监测系统，安装实时的安全监控系统，可以提高服务器系统的可靠性，使网络安全系统更加强健。选择入侵检测系统，应特别注意其主要性能的情况，包括：协议分析及检测能力；解码效率(速度)；自身安全的完备性；精确度及完整度，防欺骗能力；模式更新速度等等。

入侵检测系统是分层安全中日益被越普遍采用的成分，它将有效地提升黑客进入网络系统的门槛。入侵监测系统能够通过向管理员发出入侵或者入侵企图警告来加强当前的存取控制系统（例如防火墙）;识别防火墙通常不能识别的攻击（如来自企业内部的攻击）;在发现入侵企图之后提供必要的信息，帮助系统的移植。

入侵检测系统虽然已经被越来越被广泛地接受为有效的安全工具，但是有一点非常重要：它不能单独工作。例如，有一种工具软件可以使检测系统失效。这种被称为"Stick"的工具可发出多个有攻击表现的信息包，从而使成为其攻击目标的网络的 IDS 频繁发出警告。结果造成管理者无法分辨哪些警告是针对真正的攻击发出的，从而使 IDS 使去作用。而且，如果有攻击表现的信息包数量超过 IDS 的处理能力的话，IDS 会陷入 DoS(拒绝服务)状态。也就是说：入侵者利用 Stick 模拟大量的虚假的正面攻击来攻击网络中的入侵检测系统。这样入侵检测系统就会试图应付大量的新的攻击。这样会降低入侵检测系统的性能，如果系统不能分辨混在大量的虚假的攻击中的真正的攻击，就可能会导致系统失效。

防火墙

由于入侵检测系统的漏洞的存在，防火墙的就成为多层安全防护中必要的一层。一个防火墙为了提供稳定可靠的安全性，必须跟踪流经它的所有通信信息。为了达到控制目的，防火墙首先必须获得所有通信层和其它应用的信息，然后存储这些信息，还要能够重新获得以及控制这些信息。

防火墙仅检查独立的信息包是不够的，因为状态信息——以前的通信和其它应用信息——是控制新的通信连接的最基本的因素。对于某一通信连接，通信状态（以前的通信信息）和应用状态（其他的应用信息）是对该连接做控制决定的关键因素。因此为了保证高层的安全，防火墙必须能够访问、分析和利用通信信息、通信状态、应用状态，并做信息处理（基于以上所有元素的灵活的表达式的估算）。

安全漏洞评估系统

安全漏洞评估系统是一个漏洞和风险评估工具，用于发现、发掘和报告网络安全漏洞。一个出色的安全评估系统不仅能够检测和报告漏洞，而且还可以证明漏洞发生在什么地方以及发生的原因。它质询网络和系统；在系统间分享信息并继续探测各种漏洞直到发现所有的安全漏洞；还可以通过发掘漏洞以提供更高的可信度以确保被检测出的漏洞是真正的漏洞。这就使得风险分析更加精确并确保管理员可以把风险程度最高的漏洞放在优先考虑的位置。

最新的漏洞评估系统还采用了独特的“路径分析技术”，用以发现漏洞的根本原因。通过分析漏洞的根本原因，任何重复的漏洞、模式或异常的现象很容易被确定到是否是重要问题，或被确定到网络中的系统并且迅速被排除。

防病毒软件

防病毒软件的应用也是多层安全防护的一种必要措施。防病毒软件是专门为防止已知和未知的病毒感染你的信息系统而设计的。它的针对性很强，但是需要不断更新，而且存在一定的片面性。由于这方面的介绍已经很多，这里不作进一步的展开。

多层防护发挥的作用

让我们看看多层防护策略如何发挥作用。

即使网络中的入侵检测系统失效，防火墙、安全漏洞评估和防病毒软件还会起作用。

配置合理的防火墙能够在入侵检测系统发现之前阻止最普通的攻击。

安全漏洞评估能够发现漏洞并帮助清除这些漏洞。

如果一个系统没有安全漏洞，即使某一个攻击没有被发现，那么这样的攻击也不会成功。

即使入侵检测系统没有发现已知病毒，防火墙没能够阻止病毒，安全漏洞检测没有清除病毒传播途径，防病毒软件同样能够侦测这些病毒。

所以，在使用了多层安全防护措施以后，企图入侵你公司的信息系统的黑客要付出成数倍的代价才有可能达到入侵目的。这时，你的信息系统的安全系数得到了大大的提升。

需要特别说明的事，网络安全的多层防护并不是一个空洞的概念和设想，而是当今领先的专业厂商完全可以提供的网络安全系列产品和全面解决方案。如积极倡导并大力实践多层次防护的赛门铁克公司就扮演了一个先行者和领导者的角色。

赛门铁克企业安全系列结合了三种技术之解决方案，包括防毒、过滤解决方案与入侵防护，收购 Axent 之后，更扩展了关于这三方面之各项解决方案。在每一领域中，赛门铁克均不断开发出可以支持主要作业平台(如 Linux,Unix， Windows NT,Windows 2000, Windows Me 等)的各项技术，并在网络上布署多层防护，从防火墙、网络或电子邮件网关口，到服务器或工作站。赛门铁克完成 AXENT 收购案后，开发并推出的内容过滤、电子邮件扫描与入侵防护等应用于多层防护的一系列新技术、新产品。此外，赛门铁克企业安全系列还添加了与 IBM 合作开发完成的数字免疫系统（DIS），并正积极建立全球性的专业服务与安全咨询业务。所有这些，并结合赛门铁克著名之管理工具，包括远程遥控方案(pcAnywhere)、存储方案克隆精灵（Ghost）与中央控管中

心(System Center)，使得赛门铁克能够为企业提供网络安全多层次保护的有效策略和完整方案。

让黑客选择放弃

在结束本文前，我们再回顾一下前面提到的有关网络安全的两个基本认识：100%的安全性是不可能达到的目标；所有问题必须与将牵涉到的风险、成本及效益进行测量比对。实际上，网络安全的多层次防护正是基于这两点认识所给出的对策、方案和承诺。

最近举报率非常高的非法入侵已经很好地阐述了信息战本质上的不平等：一个专业知识有限的黑客，有时甚至是通过一次电话拨号连接，就能轻易地侵入和攻击一个企业的电脑网络，使企业直接损失上百万，并影响企业的声誉和业务开展。如果你拥有多层安全防护系统，那么，黑客渗透进来的成本就更高，他们就需要更多的资源，而这些都是大多数潜在的黑客做不到的。多层安全防护系统使得入侵者更可能放弃对你的系统的攻击。

注意论坛恶意代码

有少数违法捣蛋的人，在论坛里利用恶意的代码在别人的贴子里贴巨型的图片、屏蔽文字导致浏览上的困难，甚至是利用代码造成浏览者死机进行捣乱，严重影响了论坛和网络秩序，搞得众网虫草木皆兵。下面就以国内一个非常著名的财经论坛中一些典型的恶意代码为例，提醒各网民注意，特别是各论坛的维护管理者，请尽快修正自己系统里的技术漏洞，以免让此类事件再次发生。