白皮书

Splunk 安全性：侦测未知的恶意软件与勒索软件

使用 Windows Sysinternals 了解早期入侵迹象

splunk>
turn data into doing™

勒索软件是一种将数据当成「人质」的特定类型恶意软件，由于它本质上会破坏数据，因此对业务的伤害特别大。但是安全人员不需要为了防御勒索软件威胁而彻夜不眠。侦测出勒索软件，是从受感染网络移除遭骇装置的关键，但若想避免组织沦为恶意软件攻击的受害者，以预防为中心的完整安全防护是必要的。

本白皮书将引导使用者逐步了解如何侦测 Windows 环境中未知的恶意软件活动和早期入侵迹象。这些技术可使用 Windows Systinternals 的事件件来侦测恶意软件和勒索软件。

侦测恶意软件的难题

传统上，在 Windows 环境中侦测进阶型恶意软件或威胁入侵时，都是使用特征码型的防毒或反恶意软件产品，但是这种作法对许多人来说可能不怎么管用。大多数特征码型的反恶意软件解决方案都是使用已知特征码列表。而这就是难题所在，因为特征码型侦测在以下情况会拦截不到任何东西：

- 端点保护产品没有理想的威胁清单可侦测所有存在的特征码
- 它们无法找出端点上以新可执行文件出现的新威胁，因为没有已知的特征码可与之比较

采用传统作法的组织，将被迫必须填补处理安全漏洞，包括资料外泄、服务中断和勒索软件。这些漏洞都和无法保护及侦测端点上的活动有关。

基本上，这些组织的问题在于无法利用从 Windows 基础架构收集到的 Windows 系统活动事件。其实除了分析数据之外，我们还可以透过检视在 Windows 端点上建立的所有处理程序和会话来确认何者为正常、何者为异常。

从所有端点收集 Sysinternals 数据的难题在于它需要协调工作和适当的外部技术。此时必须在 Windows 端点安装一个可从许多 Windows 系统实时收集 Sysinternals 事件的轻量型代理程序。从端点收集到 Windows 活动的详细信息 (以事件日志格式) 之后，还必须将其储存在可处理大量讯息的数据平台中，并能够有效地搜寻及分析系统活动，以找出异常。

作法

Splunk 转寄器 (forwarder) 可让使用者从端点实时收集 Windows 基础架构的 Sysmon 资料。Splunk 软件会自动将与分析异常相关的事件传输到该端点。

Splunk 平台提供两个重要功能，可解决利用 Sysinternals 事件来侦测已知进阶型恶意软件感染的早期迹象的难题：

1. 收集 Windows 活动：Splunk 的 Windows 操作系统型转寄器可从事件日志收集所有 Sysinternals 数据
    - 提供用来收集所有 Windows 数据 (事件日志、Sysinternals、效能监控、档案) 的简单代理程序
    - 提供安全且高度可靠的传输方式，将数据集中在分析平台中
    - Sysmon 专用格式化与处理能力，可立即套用分析
2. 用来搜寻及分析异常情况的分析基础：使用简单的搜索、统计加总和计算，找出处理程序建立数据中的罕见值。
    - 透视不同的端点标准，以动态获得结果
    - 套用机器学习

透过对数据套用分析方法，Splunk 平台可让使用者去除统计计算中的正常模式，以识别出活动端点中的异常情况。这项技术可广泛使用在 1) Windows 型服务器基础架构，或 2) 用来收集所有 Windows 客户端的 Sysinternals 事件。本使用案例可运用在大多数的安全营运上。无论组织是否已拥有端点安全解决方案，这些丰富的信息皆可为端点安全评估提供重要价值，Sysinternals 甚至还可以为其他 IT 营运和服务分析提供更多相关信息。

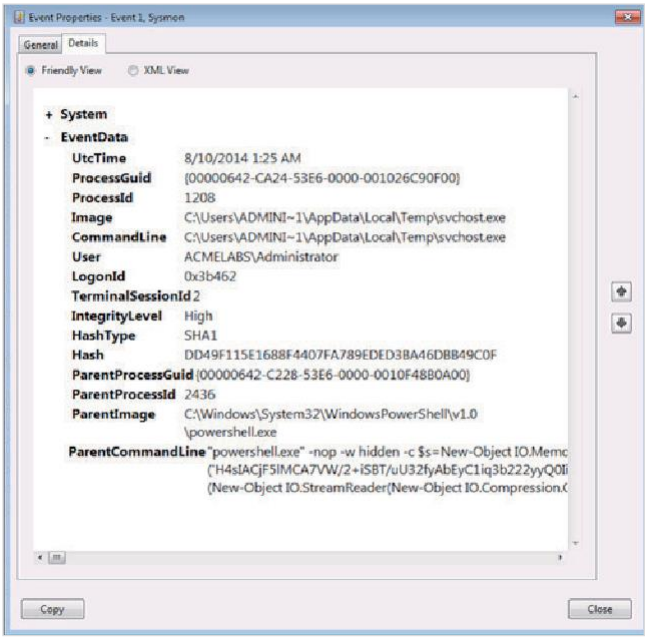
资料来源

侦测 Windows 端点上是否存在恶意软件活动所需的数据源，是 Sysinternals 使用 Sysmon 从 Windows 事件日志收集而得。组织可以安装 Microsoft 提供的 Sysmon， 然后安装 Splunk 转寄器定义出需要收集和筛选的内容， 以便取得详细信息。这些 Sysinternals 的数据是寻找特殊活动迹象的一个起点，但我们还要追踪受感染方式和内容的其他关联性，因此建议进一步撷取 Proxy、IDS/IPS、DNS/数据流数据，以根除可能感染的路径，然后确定范围并缓解事件。透过 Splunk 软件分析 Sysinternals，可以在侦测任何已知或未知的潜在恶意软件时取得明确的入侵迹象。

- Windows Sysinternals 使用 Sysmon 透过事件日志取得 (必要)
- Proxy、IDS/IPS、DNS、数据流 (当无法侦测时建议进行进一步调查)

已安装 Sysmon 的事件日志将提供以下 Splunk 软件会收集的详细数据：

- 处理程序建立，包括具有目前处理程序和父处理程序的完整命令行
- 使用 MD5、SHA1 或 SHA256 的处理程序映像的杂凑
- 处理程序 GUID，如同操作系统重复使用的 PID，可提供静态 ID，以获得更好的关联
- 两个主机之间的网络联机记录，包括 TCP/UDP的安全处理程序、IP 地址、端口号码、主机名，以及端口名称
- 档案建立时间变更
- 开机处理程序事件，可能包含内核模式恶意软件



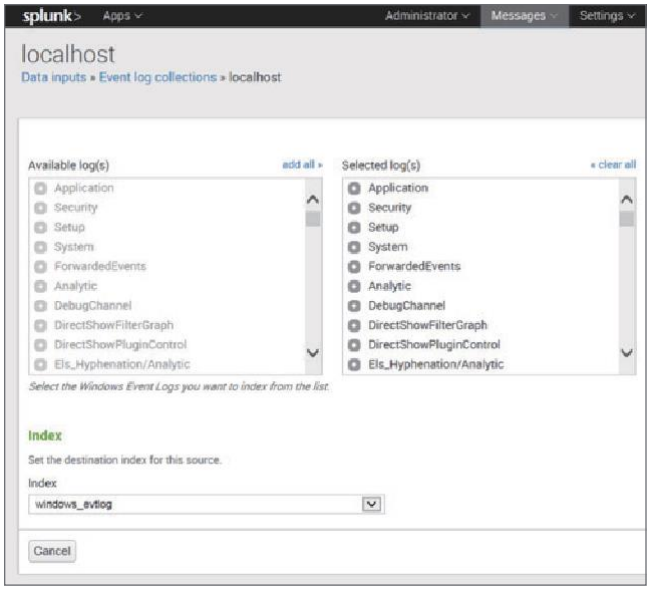
透过 Sysmon 的 Windows 事件日志的范例

收集 Windows 活动事件

使用 Splunk 转寄器从 Windows 基础架构收集各种信息片段很容易。

以下是收集 Sysmon 数据，并将其整合至 Splunk 平台的范例步骤：

1. 在 Windows 端点上安装 Sysmon (可以从下列连结下载)： https://docs.microsoft.com/en-us/sysinter- nals/ downloads/sysmon
2. 在端点安装 Splunk 转寄器，就可以将 Sysinternals 讯息实时转寄到 Splunk 实例
3. 安装适用于 Microsoft Sysmon 的 Splunk 附加组件，然后轻松地将 Splunk设定为撷取并对应至 CIM 。 在这里下载 ： https://splunk-base.splunk.com/app/1914/



一旦安装 Sysmon 之后，您就可以使用 Splunk 的「数据输入」决定您想要的内容，只要选取想传输到 Splunk 索引器 (Indexer) 的事件日志类型即可。

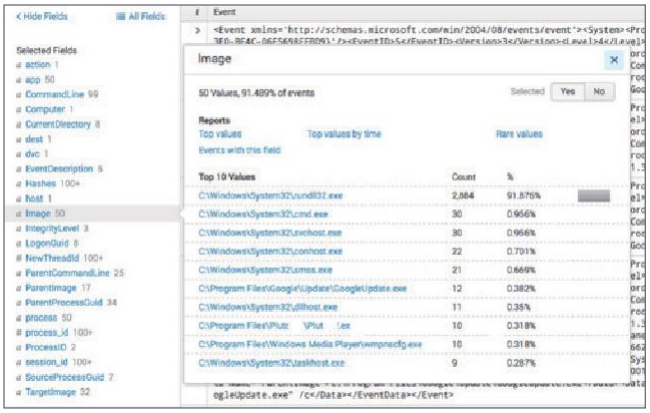
既然 Splunk 平台中有事件，就有大量的信息可供您使用。从 Splunk 索引呼叫 Sysinternals 事件的基本搜寻作法：

```
sourcetype="XmlWinEventLog:Microsoft-
Windows-Sysmon/Operational"
```

以下是在 Splunk 软件中收集数据的范例。Windows 事件日志格式会被转换成 XML，所有不同字段会整合成一行事件。

```
<Event xmlns='http://schemas.
microsoft.com/win/2004/08/events/
event'><System><Provider Name='Microsoft-
Windows-Sysmon' Guid='{5770385F-C22A-
43E0-BF4C-06F5698FFBD9}'/><EventID>1</
EventID><Version>5</Version><Level>4</
Level><Task>1</Task><Opcode>0</Opcod
e><Keywords>0x8000000000000000</
Keywords><TimeCreated SystemTime='2016-
02-04T01:58:00.125000000Z'/><EventReco
rdID>73675</EventRecordID><Correlation/
><Execution ProcessID='1664'
ThreadID='185 6'/><Channel>Microsoft-
Windows-Sysmon/ Operational</
Channel><Computer>FSAMUELS</
Computer><Security UserID='S-1-5-18'/></
System><EventData><Data
```

采用 XML 格式收集的 Sysinternals 事件数据，都会被适用于 Sysmon 的 Splunk 附加组件剖析成 Splunk 平台中的字段。现在，浏览复杂的 Sysinternals 事件容易多了，只要指向并单击剖析的字段即可。



搜寻建立异常的处理程序

挑战在于，我们如何防范未知？这里的「未知」指的是没有清单可用于验证被定义对或错，以及数据本身是否能得到正常或异常的结论。这是根据了解什么是多数，什么是少数，并与相关的其他分析信息进行关联的结果。

搜寻建立异常的处理程序

侦测活动的变化时，需要比较过去和现在发生的事件来发现异常。

用于判断异常的各种层面要件是：

- 什么是现有的，什么是新的？
- 哪些是现有、哪些是新的统计资料，以验证哪个是旧的 (正常的数据)，哪个是新的 (需要验证的数据)？
- 现有实体和新实体之间的时间关系是什么？
- 现有实体和其他实体之间的关联性，例如与其相关的资产数目。

现在我们可以利用与验证异常相关的深入信息删去正常情况，以筛选出最有可能进行评估和分析的异常情况。

将不同实体的统计数据相互比较时，很可能就会看出一些区别。

Windows Sysinternal 可提供详细数据，让我们得以了解端点在安全性和弱点方面的状态。分析 Sysinternals 的主要目的之一，就是能够了解端点上所安装和执行的处理程序与档案。有一些与处理程序执行相关的事件可显示系统上的活动，它们是重要的信息来源，可协助安全分析人员了解：

- 已经执行哪些处理程序
- 可执行文件的目录来源为何
- 执行可执行文件的父处理程序为何
- 已执行处理程序的特征为何

这些从 Sysinternals 取得的深入信息是重要的系统活动信息，在应用分析以找出端点的执行处理程序和动作是否异常时非常重要。我们可以从不同的 Sysmon 来源轻松收集到这些资料。请将 Sysmon 的哈希信息 (如 MD5、SHA1 或 SHA256) 附加到每个处理程序，以便分析人员可以识别特定系统可执行文件的不同版本。

例如，为什么我们会注意 “cmd.exe” 处理程序的完整路径？即使 “cmd.exe” 看起来是 Windows 上合法的可执行文件，但如果我们发现该二进制文件的路径奇怪，或许它会是
「害群之马」。若进一步发现该二进制文件 “cmd.exe” 的MD5 哈希与网络中其他所有 “cmd.exe” 不同，怎么办？ 代表这是档案遭窜改的明显迹象，可能会将恶意代码隐藏为合法的可执行文件。

隐藏为现有作业系统或应用程序处理程序的恶意软体处理程序

大多数计算机用户都曾经检视过 Windows 处理程序监视器，但操作系统似乎都正在执行所有正常处理程序，完全找不到问题。无论用户是谁，我们都知道计算机曾受各种恶意软件的感染。例如，「害群之马」的恶意软件会伪装成正常的操作系统处理程序。当恶意软件处理程序执行时，就如同正常处理程序般一样。如何能够侦测到这种
「害群之马」？

如果我们碰是进阶型恶意软件，例如是反恶意软件产品从未发现或侦测到的恶意软件类型，该怎么办？因为新可执行文件的特征码不明，而且这种类型的恶意软件一旦在端点上执行，将会限制大多数反恶意软件侦测并发出警示的能力。我们可以使用分析解决这一类问题吗？只要针对不同可执行文件的指纹进行一组条件分析即可。

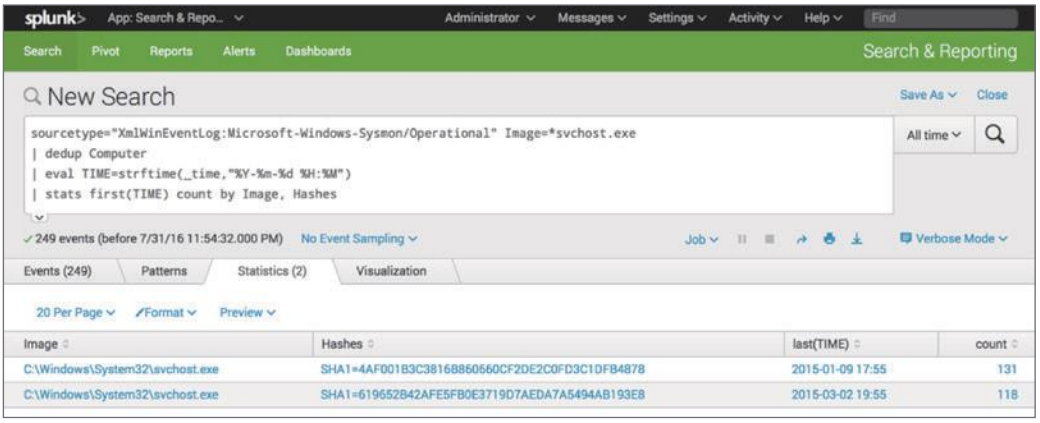
在发现这些威胁时，Sysmon 事件的哈希扮演着重要的角色。Sysmon 处理程序建立事件上附带的哈希信息，即为可执行文件的唯一指纹。如果我们将那些受信任可执行文件的现有指纹，与最近启动的类似可执行文件的新指纹进行比较，则可以找到异常的处理程序。我们可以使用可执行文件名称，透过简单的 Splunk SPL 加总来分析有关处理程序及其哈希的详细 Sysmon 事件。

无论可执行文件如何伪装自己，这种作法都可以列出可执行文件的唯一计数。哈希的指纹代表一个无可争议的唯一档案或已经执行的可执行文件。最重要的是，这些唯一的哈希计数，将指出我们需要更仔细地检视那些内容。

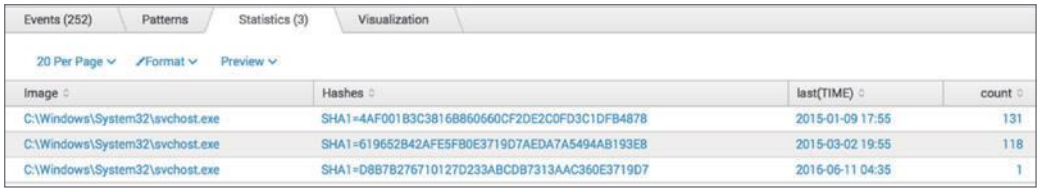
以下是搜寻语法：

```
sourcetype="XmlWinEventLog:Microsoft-Windows-Sysmon/Operational" Image=*svchost.exe
| dedup Computer
| eval TIME=strftime(_time,"%Y-%m-%d %H:%M")
| stats first(TIME) count by Image, Hashes
```

这种搜寻可寻找具有不同哈希的所有相同可执行文件名称。

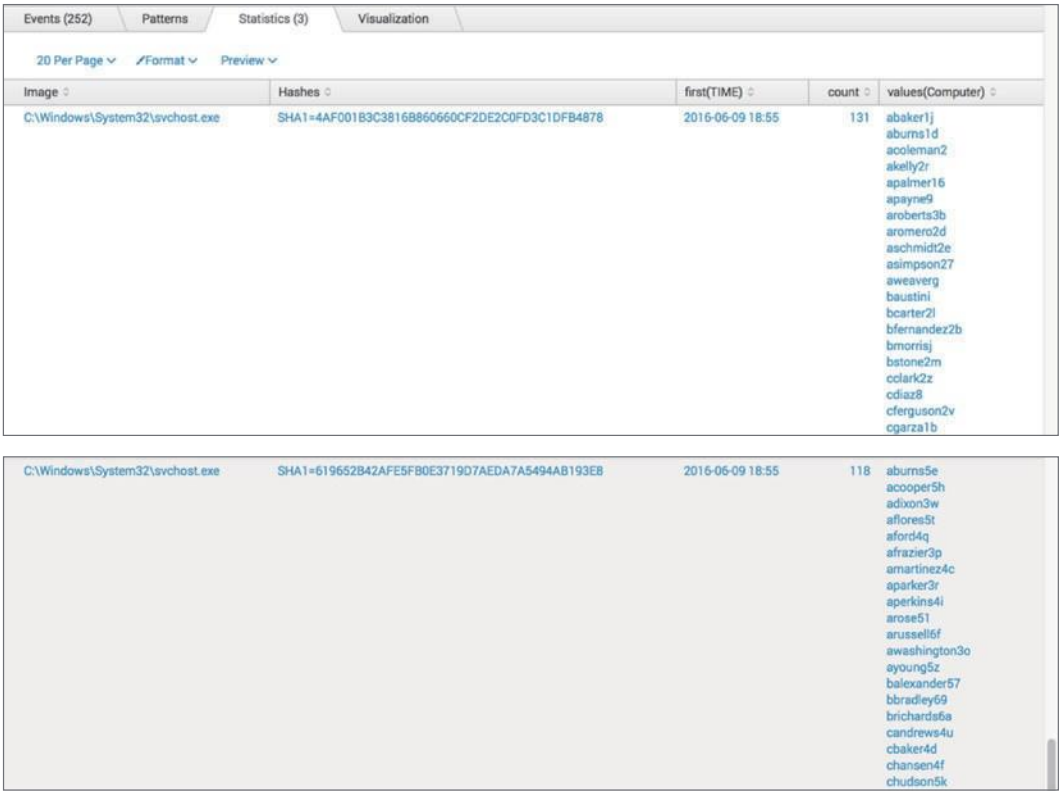


根据搜寻的结果，我们找出了路径完全相同的可执行文件 svchost.exe，但是请注意，它们的哈希是不同的。代表该档案有两种不同 Windows 操作系统的版本，因为此基础架构可以充分支持不同 Windows 版本的主机。这看起来很正常，因为网络中拥有 200 多部主机，因此关键系统处理程序 “svchosts.exe” 的哈希分布会类似于 Windows 版本的数量。请注意实例的总和，了解基础架构中执行着两个版本的操作系统，然后查看两种结果的计数，我们可以得出看起来一切正常的结论。



在下列范例中，我们得到与上例相同的搜寻回传信息。结果显示前两个多数哈希可执行文件的分布数量类似，但显示第三个哈希可执行文件的主机较少，而且发现了一个新的 SHA1 哈希。也就是说，哈希不同且处理程序数量明显较少的相同可执行文件，意味着这是一个和系统二进制文件名称相同的新可执行文件。计数总和为 “1”，表示它出现的频率很罕见，除非网络上正执行着另一个新版本的操作系统，且使用不同的系统可执行文件。若非如此，则代表它是我们需要透过 Google 搜寻参考的可疑哈希。

此外，“first(TIME)” 函数会指出该异常可执行文件首次建立的时间，并指出与正常 svchost.exe 可执行文件相比，它确定是一个全新的处理程序。first time 函数可让您深入了解现有的可执行文件与新的可执行文件，而且关联计数的总和以判断出异常可执行文件。出现次数较少的第三个哈希和时间戳较新的可执行文件，最有可能是防病毒程序侦测不到的恶意软件。



Events (252) Patterns Statistics (3) Visualization

20 Per Page Format Preview

Image	Hashes	first(TIME)	count	values(Computer)
C:\Windows\System32\svchost.exe	SHA1=4AF001B3C3816B860660CF2DE2C0FD3C1DFB4878	2016-06-09 18:55	131	abaker1j aburns1d acoleman2 akelly2r apalmer16 apayne9 aroberts3b aromero2d aschmidt2e asimpson27 aweaverg baustini bcarter2l bfernandez2b bmorrisj bstone2m cclark2z cdiaz8 cferguson2v cgarza1b
C:\Windows\System32\svchost.exe	SHA1=619652B42AFE5FB0E3719D7AEDA7A5494AB193E8	2016-06-09 18:55	118	aburns5e acooper5h adixon3w aflores5t aford4q afrazier3p amartinez4c aparker3r aperkins4i arose51 arussell6f awashington3o ayoung5z balexander57 bbradley69 brichards6a candrews4u cbaker4d chansen4f chudson5k

请务必确认哪些主机与两个不同的正常 svchost.exe 的哈希有关，以及哪些主机出现潜在的恶意软件活动。此时可以透过使用 values (Computer) 函数，从 Sysmon 数据的 “computer” 字段中列出唯一值来达成。

```
sourcetype=”XmlWinEventLog:Micro
soft-Windows-Sysmon/Operational”
Image=*svchost.exe
| dedup Computer
| eval TIME=strftime(_time,”%Y-%m-%d
%H:%M”)
| stats first(TIME), count,
values(Computer) by Image, Hashes
```

分析拥有新哈希的处理程序之后，我们可以使用几个条件来当作定义，找出潜伏在系统处理程序中的恶意软件：

- 从可执行文件的路径和名称来看，此处理程序看起来很正常，但是与现有的历史哈希相比，新可执行文件的哈希不一样。
- 与现有的可执行文件哈希相比，处理程序建立的频率显然不同。

了解这种操作的基本作法后，我们可以定义一个查询，该查询会透过计算来自动筛选，找出处理程序建立计数与现有和全新可执行文件哈希之间的数值差异。您可以使用 “eventstats” 计算所有发生次数的总和并用它来计算每个个别可执行文件出现的百分比，以轻松定义出一个相对阈值来找出「特殊可执行文件」，即使它们将自己掩饰为无害也一样。

以下是搜寻语法：

```
sourcetype=”XmlWinEventLog:Micro
soft-Windows-Sysmon/Operational”
Image=*svchost.exe
| dedup Computer
| eval TIME=strftime(_time,”%Y-%m-%d
%H:%M”)
| stats first(TIME) count by Image, Hashes
| eventstats sum(count) as total_host
| eval majority_percent=round((count/
total_host)*100,2)
```

现在，我们该如何定义搜寻 (规则)，让 Splunk 软件找出这些类型的特殊可执行文件?

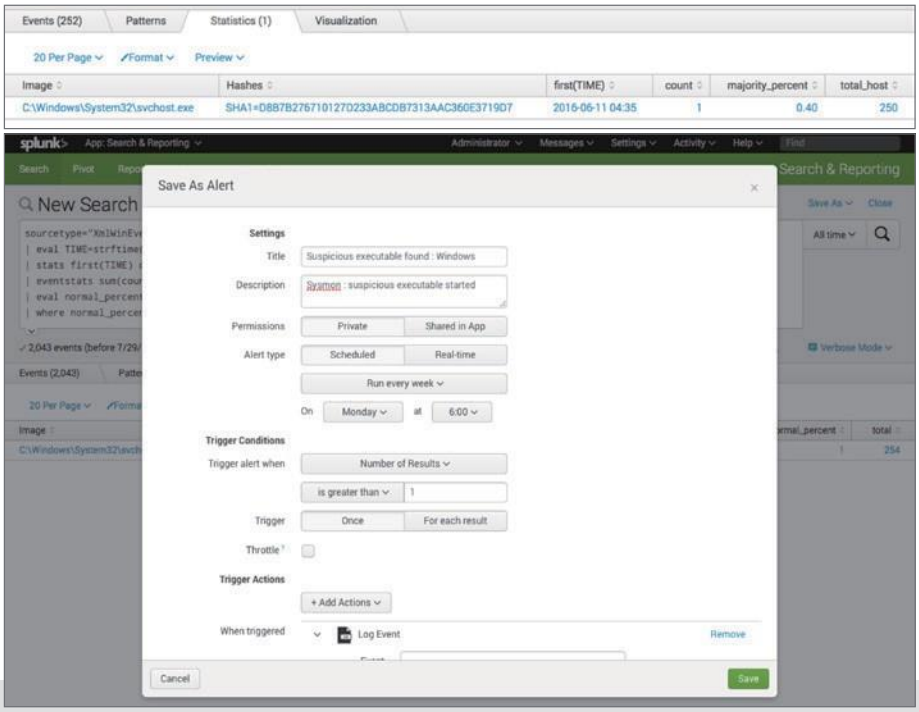
利用先前的相对计数，并使用 “majority_percent<5” 去除正常群组，然后根据相对阈值，即可找出异常的可执行文件群组。

```
sourcetype=”XmlWinEventLog:Micro
soft-Windows-Sysmon/Operational”
Image=*svchost.exe
| dedup Computer
| eval TIME=strftime(_time,”%Y-%m-%d
%H:%M”)
| stats first(TIME) count by Image, Hashes
| eventstats sum(count) as total_host
| eval majority_percent=round((count/
total_host)*100,2)
| where majority_percent<5
```

这种方法可以应用于 Splunk Enterprise 的已储存搜寻，或是 Enterprise Security 的关联搜寻功能，以便为我们进行分析，并自动向分析人员发送可能在网络上任何 Windows 工作站中出现的异常处理程序执行警示。

总结

安全分析人员可结合使用 Splunk Enterprise 和 Microsoft Sysmon 深入了解端点上的详细活动，而且能够侦测进阶型和未知的恶意软件活动。以数值来统计分析端点资料风险，能让分析人员轻松分析攻击者入侵主机的行为，还可以根据这些数值，进一步将规则定义为阈值。藉此，安全分析人员可以使用类似技术来解决仅能透过分析方法解决的问题和使用案例。这种从内容中找出差异和异常的分析方法，可让安全团队更快地侦测出进阶型威胁，并将对业务影响降至最低。



深入了解如何透过 Splunk 免费在线示范环境中的安全调查使用案例，因应恶意软件和勒索软件的威胁。