Informe Código Dañino
CCN-CERT ID-10/20

AZORult

Abril 2020

Edita:

GOBIERNO DE ESPAÑA MINISTERIO DE DEFENSA

CCN centro criptológico nacional

2.5.4.13=Qualified Certificate: AAPP-SEP-M-SW-KPSC, ou=sello electrónico, serialNumber=S2800155J, o=CENTRO CRIPTOLOGICO NACIONAL, c=ES
2020.04.22 08:25:11 +02'00'

Fecha de Edición: abril de 2020

LIMITACIÓN DE RESPONSABILIDAD

El presente documento se proporciona de acuerdo con los términos en él recogidos, rechazando expresamente cualquier tipo de garantía implícita que se pueda encontrar relacionada. En ningún caso, el Centro Criptológico Nacional puede ser considerado responsable del daño directo, indirecto, fortuito o extraordinario derivado de la utilización de la información y software que se indican incluso cuando se advierta de tal posibilidad.

AVISO LEGAL

Quedan rigurosamente prohibidas, sin la autorización escrita del Centro Criptológico Nacional, bajo las sanciones establecidas en las leyes, la reproducción parcial o total de este documento por cualquier medio o procedimiento, comprendidos la reprografía y el tratamiento informático, y la distribución de ejemplares del mismo mediante alquiler o préstamo públicos.

ÍNDICE

1. SOBRE CCN-CERT, CERT GUBERNAMENTAL NACIONAL

El CCN-CERT es la Capacidad de Respuesta a incidentes de Seguridad de la Información del Centro Criptológico Nacional, CCN, adscrito al Centro Nacional de Inteligencia, CNI. Este servicio se creó en el año 2006 como **CERT Gubernamental Nacional español** y sus funciones quedan recogidas en la Ley 11/2002 reguladora del CNI, el RD 421/2004 de regulación del CCN y en el RD 3/2010, de 8 de enero, regulador del Esquema Nacional de Seguridad (ENS), modificado por el RD 951/2015 de 23 de octubre.

Su misión, por tanto, es contribuir a la mejora de la ciberseguridad española, siendo el centro de alerta y respuesta nacional que coopere y ayude a responder de forma rápida y eficiente a los ciberataques y a afrontar de forma activa las ciberamenazas, incluyendo la coordinación a nivel público estatal de las distintas Capacidades de Respuesta a Incidentes o Centros de Operaciones de Ciberseguridad existentes.

Todo ello, con el fin último de conseguir un ciberespacio más seguro y confiable, preservando la información clasificada (tal y como recoge el art. 4. F de la Ley 11/2002) y la información sensible, defendiendo el Patrimonio Tecnológico español, formando al personal experto, aplicando políticas y procedimientos de seguridad y empleando y desarrollando las tecnologías más adecuadas a este fin.

De acuerdo a esta normativa y la Ley 40/2015 de Régimen Jurídico del Sector Público es competencia del CCN-CERT la gestión de ciberincidentes que afecten a cualquier organismo o empresa pública.

Finalmente, en el Real Decreto-ley 12/2018, de Seguridad de las Redes y Sistemas de Información, queda fijado el CCN-CERT como organismo de respuesta a incidentes al que corresponde la comunidad de referencia constituida por las entidades del ámbito subjetivo de aplicación de la Ley 40/2015, de 1 de octubre (Sector Publico, en adelante). Además, en los supuestos de especial gravedad que reglamentariamente se determinen y que requieran un nivel de coordinación superior al necesario en situaciones ordinarias, el CCN-CERT ejercerá la coordinación nacional de la respuesta técnica de los CSIRT (anteriormente denominados CERT).

2. INTRODUCCIÓN

AZORult dispone de una larga trayectoria desde 2016, donde fue utilizado como segundo *stage* por parte del troyano bancario Chthonic [REF.- 1]. La versión 2 de este troyano, desarrollada en Delphi incorporó, entre otros [REF.- 2], soporte para dominios ".bit" facilitando el anonimato y dificultando así el bloqueo de sus servidores de control. Además, implementaba funcionalidades para robar: *cookies*, credenciales almacenadas en un gran número de aplicaciones (navegadores, clientes de mensajería y correo, etc.) y múltiples tipos de criptomonedas (electrum, monero, bitcoin, etc.).

Este código dañino también implementa funcionalidades de *downloader* pudiendo descargar nuevos especímenes en el sistema comprometido. En julio de 2018, investigadores de Proofpoint detallaban una campaña atribuida al grupo TA516 [REF.- 3] en donde, por medio de una macro en un documento Word, se descargaba y ejecutaba una instancia de AZORult que, a su vez, tras exfiltrar las credenciales de la víctima, servía de *dropper* para el *ransomware* Hermes. Asimismo, investigadores de Salesfoce identificaban ese mismo mes una campaña [REF.- 4] asociada al grupo de atacantes Oktropys en donde una variante de AZORult desembocaba en la instalación del *ransomware* Aurora.

En octubre de 2018 el principal vendedor de AZORult, el usuario CrydBrox, anunciaba en el famoso foro ruso *exploit.in* [REF.- 5] la venta de la versión v3.3 destacando mejoras significativas: menos ratio de detección, soporte de extracción de nuevas criptomonedas, mejoras en su *loader*, etc.

Es posible que, a raíz del *leak* del código de AZORult por parte de Check Point Research [REF.- 6] en septiembre de 2018 (versiones 3.1 y 3.2), CrydBrox decidiera anunciar a finales de ese mismo año el cese de su venta. Estos hechos posiblemente incentivaron también el desarrollo de nuevos *builders* de AZORult, como "Gazorp" [REF.- 7], poco después que se filtrara su código fuente.

El *leak* tanto de su *builder* como del panel de control ha propiciado desde entonces el desarrollo de nuevas versiones de AZORult permitiendo que grupos de atacantes de todo tipo implementen sus propias versiones de este *código dañino*. En marzo de 2019, por ejemplo, SecureList identificaba una versión en C++ de AZORult [REF.- 8]. A pesar de que esta nueva versión presenta ciertas deficiencias respecto de su predecesor en Delphi AZORult 3.3 (por ejemplo, carece de *loader* y de soporte para extraer credenciales de diversos navegadores) implementa nuevas funcionalidades que lo hacen más peligroso (por ejemplo, la activación de escritorio remoto en el equipo).

El código dañino conocido como Ramnit o Emotet han sido empleados también para descargar AZORult por medio de malspam, phishing y exploits Kits como RIG [REF.-9]. Durante el último año se han identificado diversas campañas en donde AZORult ha formado parte del arsenal de herramientas empleado por los atacantes. Por ejemplo, en abril de 2020 el equipo de analistas de Talos [REF.- 10] identificaba una campaña, con fines principalmente económicos, en donde el grupo de actores utilizaban AZORult junto con el *miner* de monero XMRigCC, una variante del RAT Remcos y el backdoor DarkVNC.

3. RESUMEN EJECUTIVO

La muestra analizada se corresponde con AZORult, un *stealer* multipropósito ampliamente vendido en determinados foros *underground* hasta 2018 y utilizado desde hace años en incidentes relacionados principalmente con el robo de información bancaria (tarjetas de crédito, carteras Bitcoin, etc.) y credenciales.

Recientemente ha sido encontrado diseñado especialmente para aprovecharse de la preocupación por el **COVID-19; en concreto, se está distribuyendo haciéndose pasar por un "mapa del coronavirus"**. El mapa se distribuye como una app, como una alternativa más fiable para rastrear la evolución del COVID-19; sin embargo, en realidad este ejecutable oculta una evolución de AZORult.

La versión analizada en el presente informe forma parte de una campaña iniciada en noviembre de 2019. El binario finge corresponderse con un cliente legítimo de VPN (ProtonVPN) que es descargado desde el dominio dañino *protonvpn.store* por medio de técnicas de *malvertising* [REF.- 11]. Dicho dominio es un *clon* del dominio legítimo ProtonVPN (https://protonvpn.com/) perteneciente a *Proton Technologies AG* (sociedad responsable también del archiconocido servicio de mail ProtonMail). Cabe destacar que no es la primera vez que atacantes utilizan sitios VPN falsos para propagar este espécimen. En mayo de 2019 un supuesto cliente VPN denominado Pirate Chick [REF.- 12] era utilizado también para servir, entre otros, AZORult.

El código dañino está protegido con el *packer* Armadillo lo que dificulta en gran medida su análisis. Cuando comienza su ejecución recopila determinada información sobre el sistema comprometido y se la remite al servidor de control vía HTTP. Posteriormente comienza con la extracción de credenciales de determinadas aplicaciones y servicios (clientes de FTP, correo, navegadores, servicios de mensajería, etc.) así como las carteras ligadas a determinadas criptomonedas.

4. CARACTERÍSTICAS DEL CÓDIGO DAÑINO

El código dañino realiza las siguientes acciones:

- Finge corresponderse con un cliente VPN legítimo.
- Recopila información del sistema y se la transmite al C2 vía HTTP de forma ofuscada.
- Extrae cookies, credenciales de múltiples aplicaciones y servicios, así como carteras asociadas a diversas criptomonedas.

Según la matriz Mitre ATT&CK, este código dañino está actualmente relacionado con un total de 16 técnicas dentro del dominio *enterprise* [REF.- 13]:

Domain	ID	Name	Use
Enterprise	T1134	Access Token Manipulation	Azorult can call WTSQueryUserToken and CreateProcessAsUser to start a new process with local system privileges.[1]
Enterprise	T1503	Credentials from Web Browsers	Azorult can steal credentials from the victim's browser.[1]
Enterprise	T1081	Credentials in Files	Azorult can steal credentials in files belonging to common software such as Skype, Telegram, and Steam.[1]
Enterprise	T1140	Deobfuscate/Decode Files or Information	Azorult uses an XOR key to decrypt content and uses Base64 to decode the C2 address.[1][2]
Enterprise	T1083	File and Directory Discovery	Azorult can recursively search for files in folders and collects files from the desktop with certain extensions.[1]
Enterprise	T1107	File Deletion	Azorult can delete files from victim machines.[1]
Enterprise	T1057	Process Discovery	Azorult can collect a list of running processes by calling CreateToolhelp32Snapshot.[1][2]
Enterprise	T1093	Process Hollowing	Azorult can decrypt the payload into memory, create a new suspended process of itself, then inject a decrypted payload to the new process and resume new process execution.[1]
Enterprise	T1012	Query Registry	Azorult can check for installed software on the system under the Registry key `Software\Microsoft\Windows\CurrentVersion\Uninstall`.[1]
Enterprise	T1105	Remote File Copy	Azorult can download and execute additional files. Azorult has also downloaded a ransomware payload called Hermes.[1][2]
Enterprise	T1113	Screen Capture	Azorult can capture screenshots of the victim's machines.[1]
Enterprise	T1032	Standard Cryptographic Protocol	Azorult can encrypt C2 traffic using XOR.[1][2]
Enterprise	T1082	System Information Discovery	Azorult can collect the machine information, system architecture, the OS version, computer name, Windows product name, the number of CPU cores, video card information, and the system language.[1][2]
Enterprise	T1016	System Network Configuration Discovery	Azorult can collect host IP information from the victim's machine.[1]
Enterprise	T1033	System Owner/User Discovery	Azorult can collect the username from the victim's machine.[1]
Enterprise	T1124	System Time Discovery	Azorult can collect the time zone information from the system.[1][2]

Figura 1. Mitre ATT&CK: *AZORult*

5. DETALLES GENERALES

La firma del código dañino es la siguiente:

Fichero	SHA1
ProtonVPN_win_v1.10.0.exe	1E7DFE574093BA983BE1B51FF5D433C16D86E072

La muestra fue vista por primera vez el 26 de noviembre de 2019 y, a fecha de realización del presente informe, presenta una ratio de detección muy alta de soluciones antivirus.

Existen indicios, a partir de determinadas firmas estáticas tales como el *linker version* (SR = *Silicon Realms*) y el nombre de determinadas secciones, que el binario ha sido empaquetado con Armadillo.

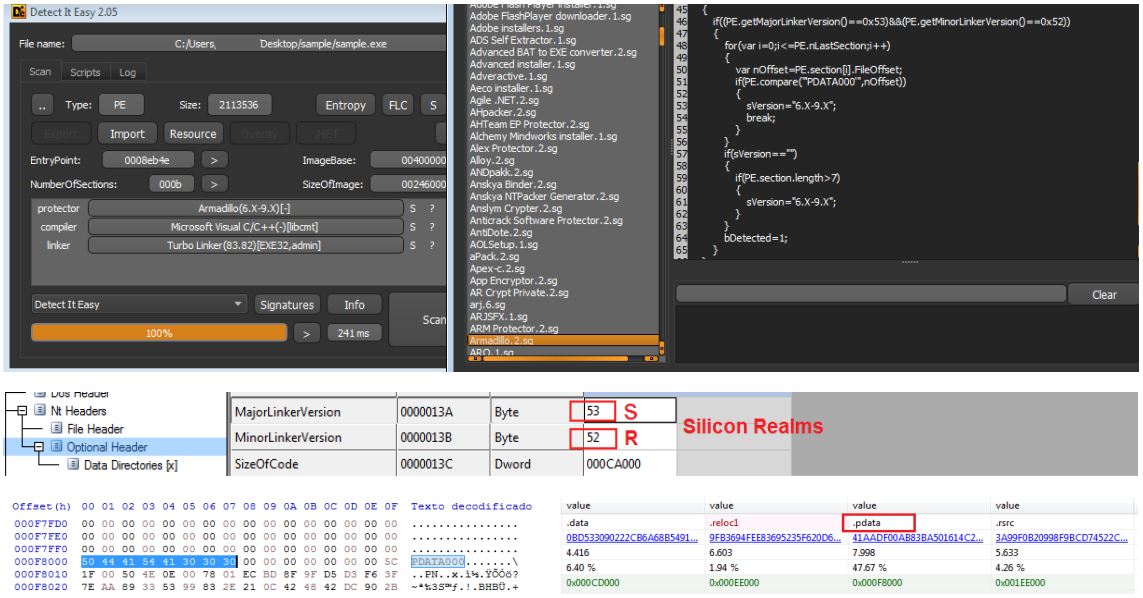


Figura 2. Packer Armadillo

El binario está compilado para arquitecturas de 32 bits y presenta como fecha de compilación en su *timestamp* el 19 de junio de 1992 (fecha comúnmente establecida por el *linker* de versiones antiguas de Delphi; por ejemplo, Delphi 5).

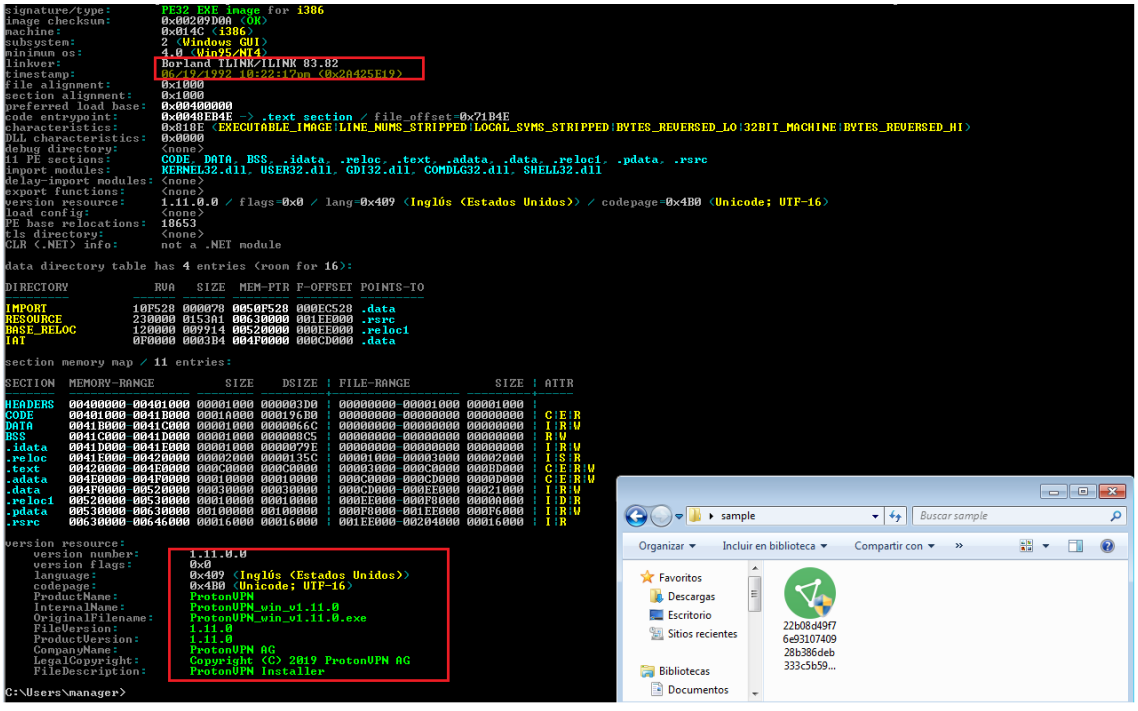


Figura 3. Información PE / Propiedades

Fíjese que el binario cuenta con el logo original del cliente legítimo de VPN ProtonVPN y que, en sus propiedades, también se referencia al mismo para dotar de mayor credibilidad al ejecutable.

Para clonar el sitio de descarga legítimo (*https://protonvpn.com/*) los atacantes han utilizado el *crawler open-source* HTTrack tal y como revela el siguiente *snapshot*

del 20 de febrero indexado por *web.archive.org* (actualmente el dominio dañino *protonvpn.store* no está disponible).

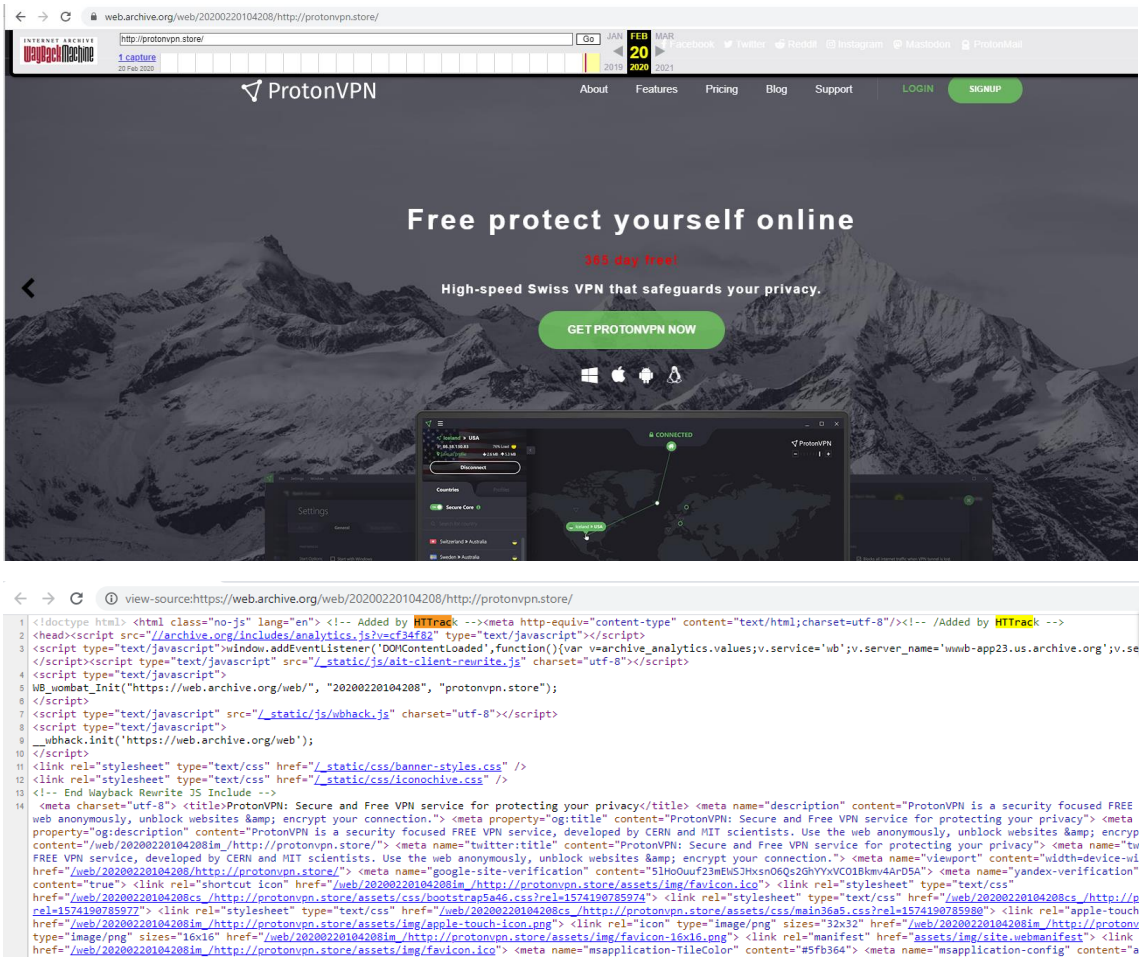


Figura 4. Dominio dañino *prontonvpn.store*

Una búsqueda de la muestra, revela que realiza conexiones con diversas IP (pertenecientes a proveedores rusos) vinculadas también con otras muestras de AZORult en los últimos meses.

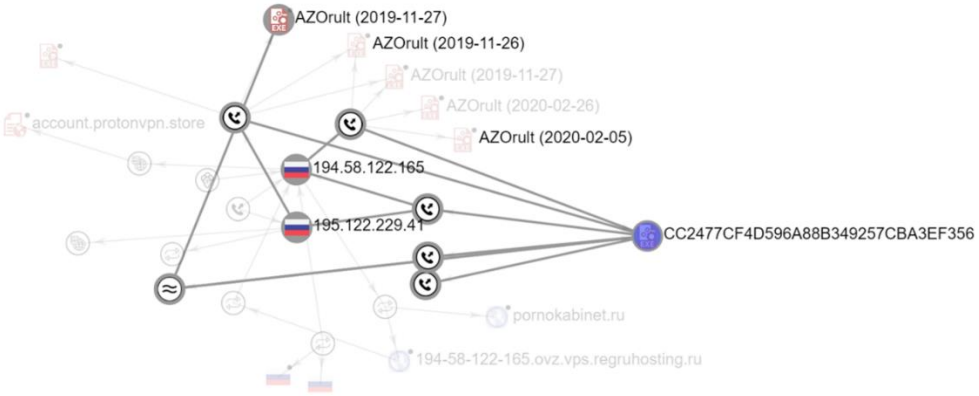


Figura 5. Vínculos de AZORult

Cabe destacar también que el binario dispone del valor *requireAdministrator* en la directiva *requestExecutionLevel* de su fichero de manifiesto para forzar la ejecución del binario con permisos administrativos.

Figura 6. Permisos administrativos

6. PROCEDIMIENTO DE INFECCIÓN

Según investigaciones de SecureList [REF.- 11] al menos una de las vías de infección de esta campaña ha sido mediante redes de *banners* de afiliación (*malvertising)*.

7. CARACTERÍSTICAS TÉCNICAS

El código dañino intentará obtener un *handle* a un objeto mutex cuyo nombre deriva del PID del proceso.

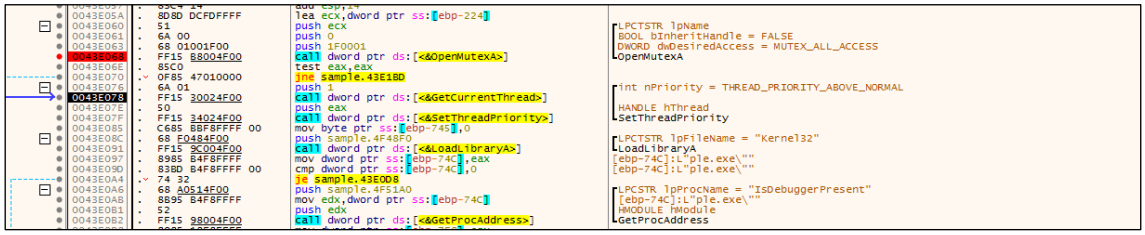


Figura 7. Comprobación mutex

Posteriormente asigna al *thread* actual una prioridad de THREAD_PRIORITY_ABOVE_NORMAL por medio de la API SetThreadPriority y comprueba si el código se está ejecutando en un *debugger* a través de una llamada a *IsDebugPresent().*

Se observa que algunas partes del código han sido diseñadas para ejecutarse solo en entornos Windows 9x. Por ejemplo, realiza una llamada a *RegisterServiceProcess* (comúnmente utilizada por código dañino para ocultarse del *task-list*). Esta API fue eliminada de NT y no funciona en XP y versiones superiores. Posiblemente dicha funcionalidad forme parte del *packer* Armadillo, empleado para ofuscar la muestra actual.

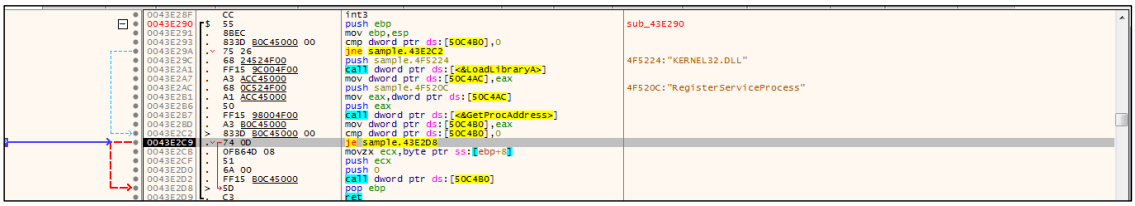


Figura 8. RegisterServiceProcess

El código dañino llevará acabo *self-debugging* para dificultar la labor de *reversing* (una de las técnicas conocidas como *Debug Blocker* implementada en Armadillo). Para ello, en primer lugar, creará un proceso hijo de sí mismo mediante *CreateProcess* utilizando como *flag* DEBUG_PROCESS.

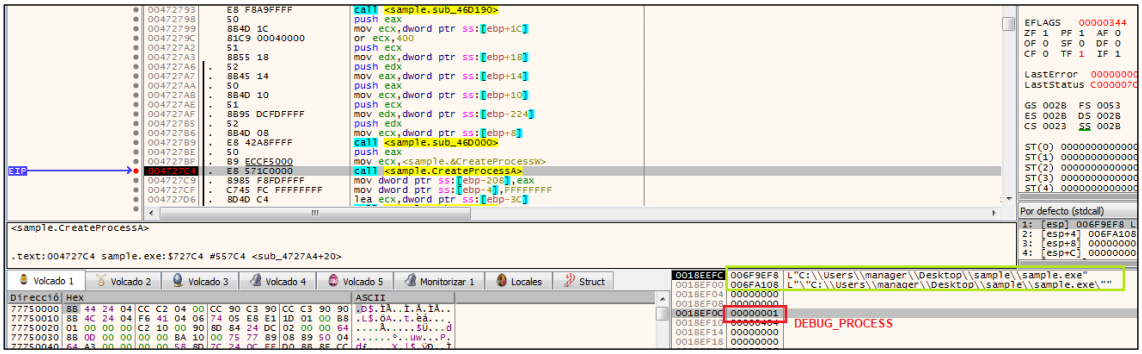


Figura 9. CreateProcessA (DEBUG_PROCESS)

El *unpacking stub* funcionará a modo de *debugger* (proceso padre) de su proceso hijo (*debugee*) y controlará las acciones del mismo por medio de eventos de *debugging*. Debido a que el proceso hijo ya está siendo depurado no es viable hacer un *attach* al mismo mediante "*DebugActiveProcess()*".

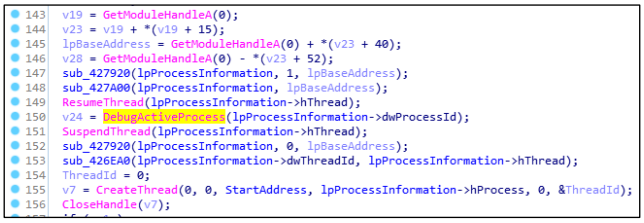


Figura 10. DebugActiveProcess

Únicamente el proceso de *unpacking* se ejecutará en el proceso hijo coordinado desde el proceso padre mediante eventos de *debugging* y APIs como *WaitForDebugEvent/ContinueDebugEvent*. En la siguiente imagen se muestra la función encargada de dicha gestión.

```
while ( (unsigned __int8)check_exitCode(lpProcessInformation->hProcess) )
{
  if ( WaitForDebugEvent(lpDebugEvent, 0x3E8u) )
  {
    if ( v104 && *((_DWORD *)::lpBaseAddress + 8) )
      v104 = 0;
    EnterCriticalSection(&CriticalSection);
    if ( lpDebugEvent->dwProcessId == lpProcessInformation->dwProcessId )
    {
      if ( !byte_50C1B3
        && v103
        && dword_50C1C8
        && (*(_DWORD *)(dword_50C1FC + 100) ^ *(_DWORD *)(dword_50C1FC + 108) ^ *(_DWORD *)(dword_50C1FC + 72)) & 8
        && GetTickCount() - dword_50C1C8 > 0x493E0 )
      {
        v103 = 0;
      }
      dwContinueStatus = -2147418111;
      switch ( lpDebugEvent->dwDebugEventCode )
      {
        case 1u:
          v99 = *(_DWORD *)(dword_50C1FC + 8) ^ lpDebugEvent->u.Exception.ExceptionRecord.ExceptionCode;
          if ( byte_50C1AD && v99 == (*(_DWORD *)(dword_50C1FC + 8) ^ 0x80000001) )
          {
            v95 = lpDebugEvent->u.Exception.ExceptionRecord.ExceptionInformation[1];
            if ( !dword_50C1D4 )
            {
              memset(&Context, 0, 0x2CCu);
              Context.ContextFlags = 65537;
              if ( GetThreadContext(lpProcessInformation->hThread, &Context) )
              {
                v93 = (v95 - dword_50C1BC) >> 12;
                if ( v93 >= 0 && v93 < dword_50C1D0 )
                {
                  v92 = (Context.Eip - dword_50C1BC) >> 12;
                  if ( v93 != v92 )
                    dword_50C1BC += dword_50C1D0 << 12;
                }
              }
```

Figura 11. Armadillo packer (gestión de eventos de *debugging*)

El proceso hijo, una vez desempaquetado parcialmente, dispone en su espacio de direcciones de todas las DLLs necesarias. Además, se pueden observar, por ejemplo, *strings* en claro que dan pistas sobre algunas de las acciones del código dañino (*headers* HTTP, API utilizadas, entradas de registro, versiones de software, etc.). En la siguiente imagen se aprecia, por ejemplo, el C2 empleado, así como evidencias que corroboran el uso del packer Armadillo.

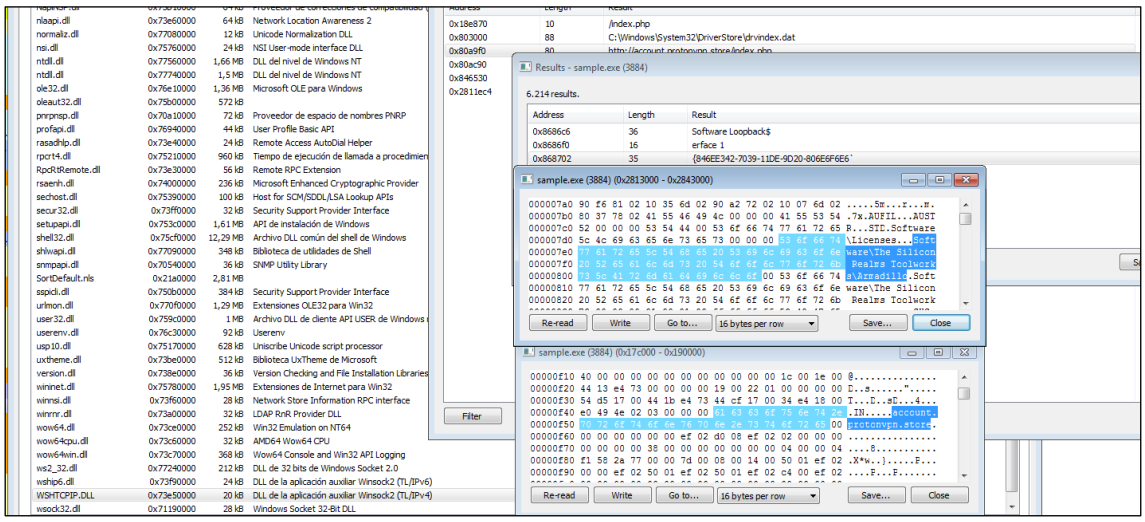


Figura 12. Espacio de direcciones del proceso hijo

Es importante destacar que Armadillo implementa *Shifting-decode-frames* o *partial code revelation* [REF.- 14]; funcionalidad que permite descifrar únicamente las funciones demandadas.

Los permisos de determinadas páginas son modificados para generar excepciones cuando son ejecutadas. Estas excepciones serán capturadas por el proceso padre quien escribirá en su espacio de direcciones vía *WriteProcessMemory*. Tras modificar las mismas reanudará la ejecución del proceso hijo hasta que alcance

otra página protegida. Para más información técnica sobre algunas de las defensas implementadas en Armadillo (*CopyMem 2 + Debug Blocker + IAT Elimination + Code Splicing + Nanomite*) remítase a la referencia adjunta [REF.-15]. Esta técnica dificulta en gran medida la ingeniería inversa al ocultar aquellas funciones que no forman parte del flujo de ejecución del proceso.

El código dañino intentará robar credenciales y cookies guardadas en un gran número de navegadores [REF.- 15]. Asimismo, implementa funcionalidades para extraer información de interés (en su mayoría credenciales, *wallets,* etc.) de un gran listado de aplicaciones entre los que se encuentran servicios de mensajería como Skype o Telegram.

Figura 13. Extracción de credenciales

8. CIFRADO Y COMUNICACIONES

El código dañino intentará comunicarse con el servidor de control mediante HTTP (por medio de la API WinInet) para remitir información del sistema y obtener las instrucciones pertinentes. Dicha información es enviada vía POST de forma ofuscada. Las conexiones con el servidor de control (*account.protonvpn.store*) se ejecutarán desde el proceso hijo (PID 932 en la imagen).

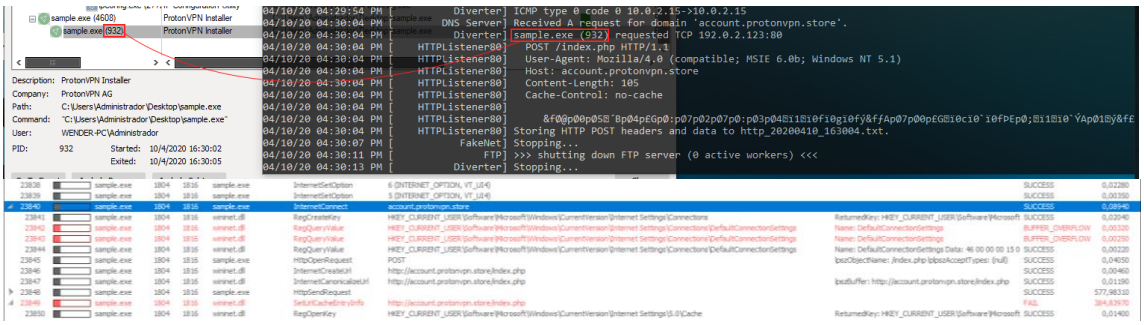


Figura 14. Petición POST (WinInet API)

Dicho POST remitirá el bot-ID [REF.-16] asociado a la víctima el cual se genera a partir de 5 campos de la máquina: *MachineUID,* Product Name, User Name, el Host Name y un *string* concatenando los valores anteriores. De cada uno de estos 5 campos se genera un ID de 4 bytes que posteriormente se codifica en URL *encode*.

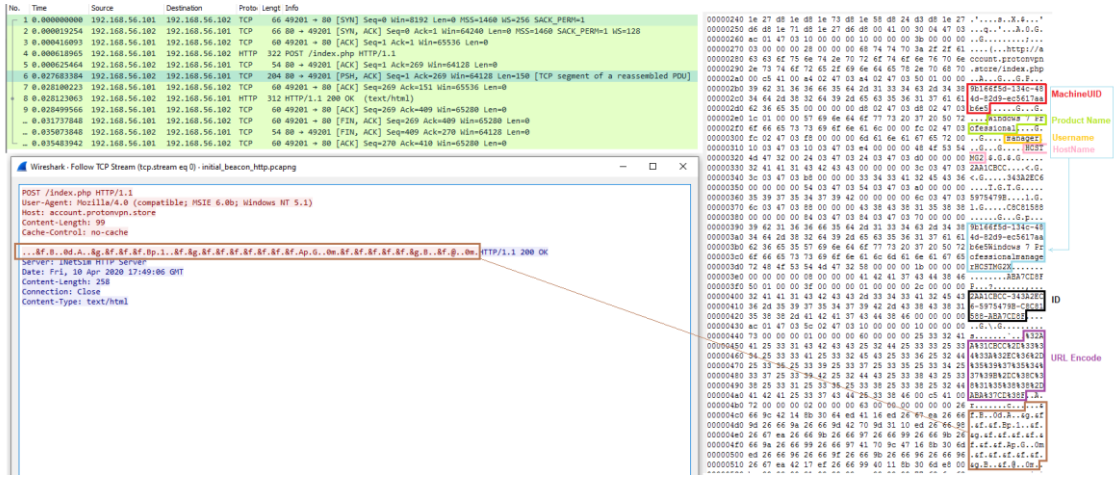


Figura 15. Envío Bot-ID

El código dañino también recupera información de la IP pública de la víctima haciendo uso del servicio *http://ip-api.com/json*

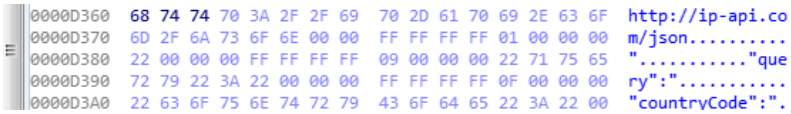


Figura 16. Comprobación Country-Code (ip-api.com)

9. PERSISTENCIA

No se ha identificado persistencia; no obstante, AZORult dispone de funcionalidades para descargar y ejecutar nuevos binarios. Es posible, por tanto, que el C2 instruya al espécimen a ejecutar nuevos binarios en el sistema los cuales sí implementen persistencia.

10. DETECCIÓN Y ELIMINACIÓN

Utilizando las reglas de Yara y Snort adjuntas en el informe se podrá detectar la presencia de AZORult. Teniendo en cuenta que el dominio *protonvpn.store* será utilizado para descargar el binario dañino y para comunicarse con el servidor de control (*account.protonvpn.store*), este indicador será de gran importancia para identificar de forma temprana esta amenaza. Asimismo, consúltese y actualice sus IDS con los dominios reportados en *https://azorult-tracker.net* y http://cybercrime-tracker.net/index.php?s=0&m=40&search=AZORult

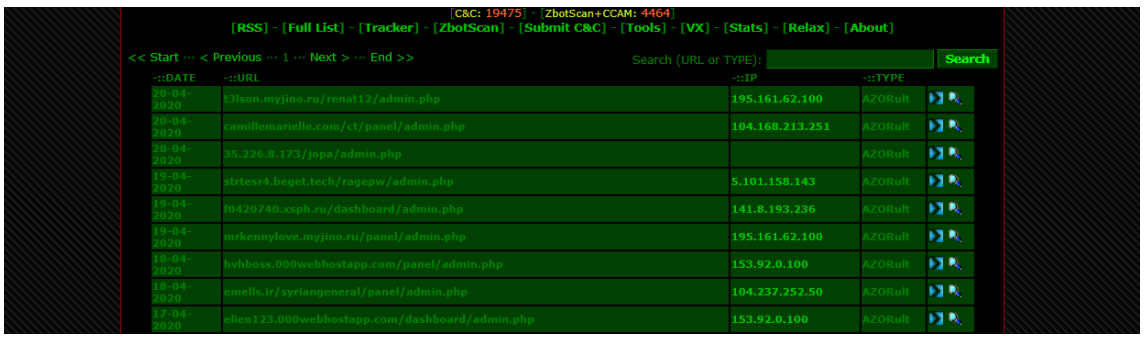


Figura 17. C2 asociados a AZORult (CyberCrime)

11. INFORMACIÓN DEL SERVIDOR DE CONTROL

El binario utilizará como servidor de control el dominio *account.protonvpn.store*.

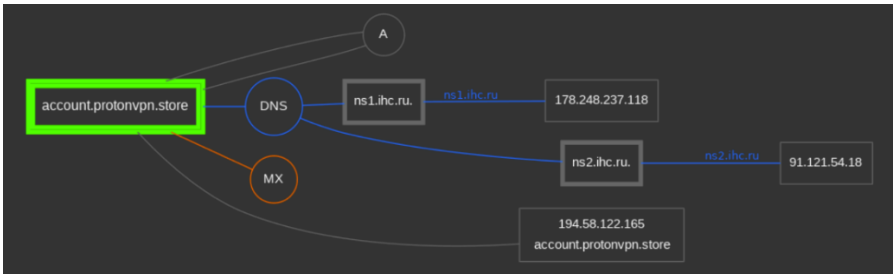


Figura 18. Dominio *account.protonvpn.store*

Este dominio fue registrado el 26 de noviembre de 2019 y desde entonces ha estado vinculado con tres IPs: 195.122.229.41, 194.58.122.165, 91.218.229.12 (todas pertenecen a proveedores de Rusia tal y como muestra su información de whois).

Date resolved	IP
2020-02-18	91.218.229.12
2020-01-20	194.58.122.165
2019-12-11	195.122.229.41

Figura 19. DNS Pasivo

Se puede inferir, currelando información de las resoluciones de IP, junto con el tipo de código dañino descargado desde cada IP, que los equipos 195.122.229.41 y 194.58.122.165 son los involucrados con las campañas de AZORult.

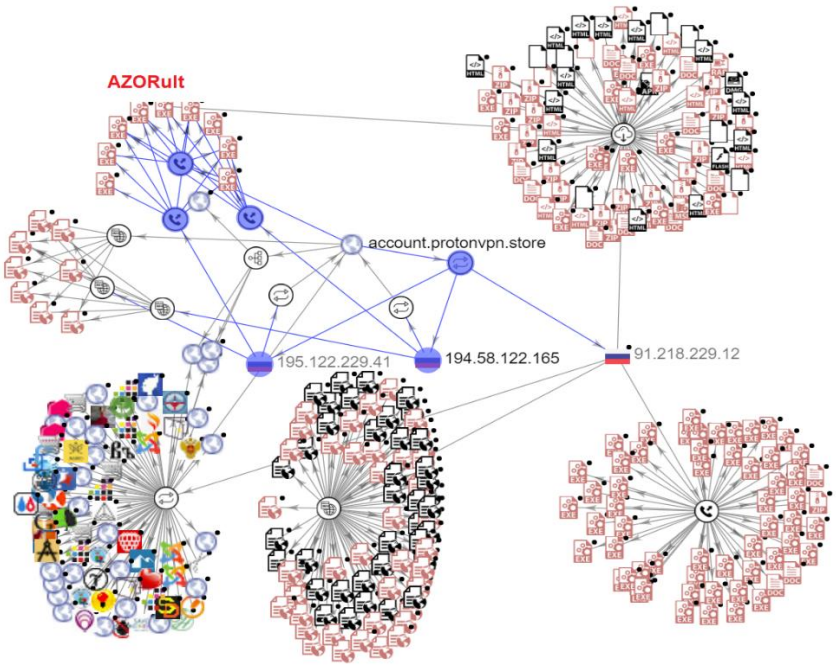
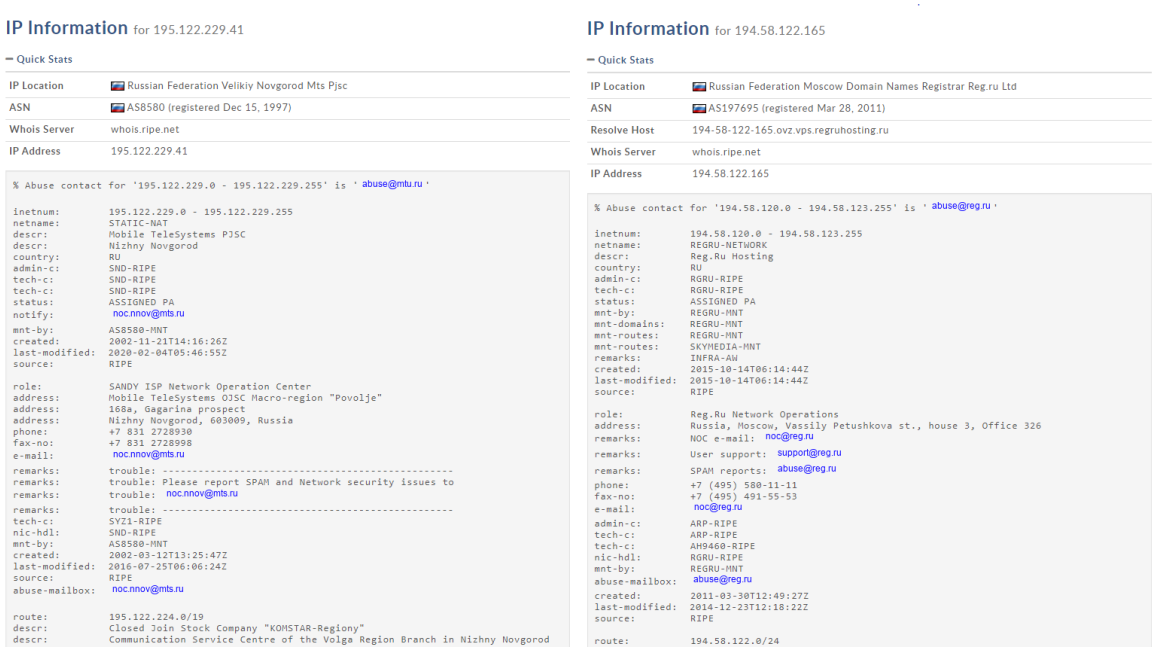


Figura 20. Direcciones IP relacionadas con AZORult

La información de *whois* asociada a cada IP se muestra a continuación:



IP Information for 195.122.229.41

Quick Stats

IP Location	Russian Federation Velikiy Novgorod Mts Pjsc
ASN	AS8580 (registered Dec 15, 1997)
Whois Server	whois.ripe.net
IP Address	195.122.229.41

```
% Abuse contact for '195.122.229.0 - 195.122.229.255' is ' abuse@mtu.ru '

inetnum:        195.122.229.0 - 195.122.229.255
netname:        STATIC-NAT
descr:          Mobile TeleSystems PJSC
descr:          Nizhny Novgorod
country:        RU
admin-c:        SND-RIPE
tech-c:         SND-RIPE
tech-c:         SND-RIPE
status:         ASSIGNED PA
notify:         noc.nnov@mts.ru
mnt-by:         AS8580-MNT
created:        2002-11-21T14:16:26Z
last-modified:  2020-02-04T05:46:55Z
source:         RIPE

role:           SANDY ISP Network Operation Center
address:        Mobile TeleSystems OJSC Macro-region "Povolje"
address:        168a, Gagarina prospect
address:        Nizhny Novgorod, 603009, Russia
phone:          +7 831 2728930
fax-no:         +7 831 2728998
e-mail:         noc.nnov@mts.ru
remarks:        trouble: ------------------------------------------------
remarks:        trouble: Please report SPAM and Network security issues to
remarks:        trouble:  noc.nnov@mts.ru
remarks:        trouble: ------------------------------------------------
tech-c:         SYZ1-RIPE
nic-hdl:        SND-RIPE
mnt-by:         AS8580-MNT
created:        2002-03-12T13:25:47Z
last-modified:  2016-07-25T06:06:24Z
source:         RIPE
abuse-mailbox:  noc.nnov@mts.ru

route:          195.122.224.0/19
descr:          Closed Join Stock Company "KOMSTAR-Regiony"
descr:          Communication Service Centre of the Volga Region Branch in Nizhny Novgorod
```

IP Information for 194.58.122.165

Quick Stats

IP Location	Russian Federation Moscow Domain Names Registrar Reg.ru Ltd
ASN	AS197695 (registered Mar 28, 2011)
Resolve Host	194-58-122-165.ovz.vps.regruhosting.ru
Whois Server	whois.ripe.net
IP Address	194.58.122.165

```
% Abuse contact for '194.58.120.0 - 194.58.123.255' is ' abuse@reg.ru '

inetnum:        194.58.120.0 - 194.58.123.255
netname:        REGRU-NETWORK
descr:          Reg.Ru Hosting
country:        RU
admin-c:        RGRU-RIPE
tech-c:         RGRU-RIPE
status:         ASSIGNED PA
mnt-by:         REGRU-MNT
mnt-domains:    REGRU-MNT
mnt-routes:     REGRU-MNT
mnt-routes:     SKYMEDIA-MNT
remarks:        INFRA-AW
created:        2015-10-14T06:14:44Z
last-modified:  2015-10-14T06:14:44Z
source:         RIPE

role:           Reg.Ru Network Operations
address:        Russia, Moscow, Vassily Petushkova st., house 3, Office 326
remarks:        NOC e-mail:  noc@reg.ru
remarks:        User support:  support@reg.ru
remarks:        SPAM reports:  abuse@reg.ru
phone:          +7 (495) 580-11-11
fax-no:         +7 (495) 491-55-53
e-mail:         noc@reg.ru
admin-c:        ARP-RIPE
tech-c:         ARP-RIPE
tech-c:         AH9460-RIPE
nic-hdl:        RGRU-RIPE
mnt-by:         REGRU-MNT
abuse-mailbox:  abuse@reg.ru
created:        2011-03-30T12:49:27Z
last-modified:  2014-12-23T12:18:22Z
source:         RIPE

route:          194.58.122.0/24
```

Figura 21. Información Whois

El servicio *https://azorult-tracker.net/,* creado específicamente para monitorizar los C2 asociados a la familia AZORult (en concreto de las versiones 3.2 and 3.3.1), revela que la IP 194.58.122.165 ha estado involucrada con múltiples víctimas procedentes principalmente de EEUU, Irlanda y Reino Unido. La gráfica ofrecida por este servicio permite mostrar los rangos de fecha con mayor actividad. Fíjese que también se proporciona la URL asociada al panel de administración: *http://account.protonvpn.store/panelka/admin.php*

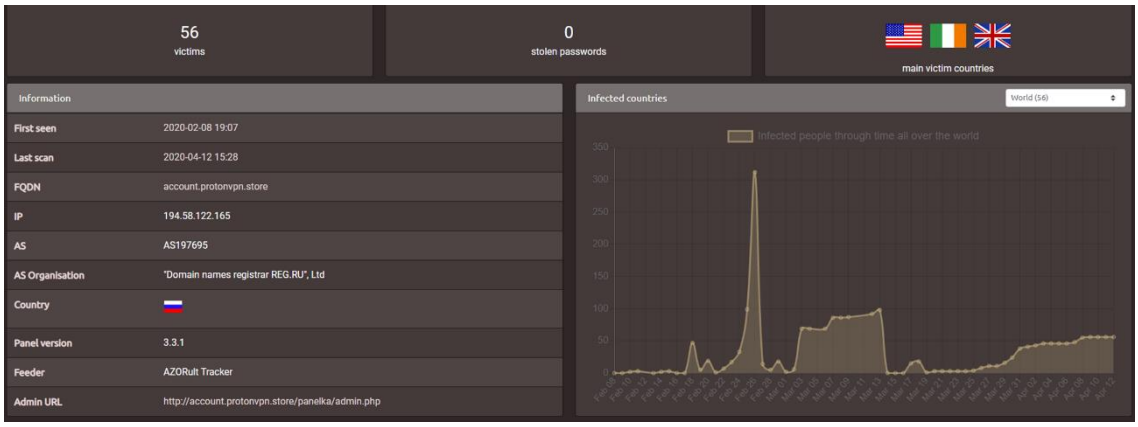


Figura 22. Información: azorult-tracker.net

12. REGLAS DE DETECCIÓN

12.1 REGLA DE YARA

Utilícese la siguiente regla de Yara para identificar el binario principal de infección utilizado en la campaña descrita.

```
import "pe"
rule azorult_protonvpn{
meta:
        description = "Azorult ProtonVPN packed with Armadillo"
        author = "CCN-CERT"
        version = "1.0"
        date = "2020-04-12"
        hash1 = "1E7DFE574093BA983BE1B51FF5D433C16D86E072"

strings:
        $x1 = "ProtonVPN" nocase wide

condition:
        uint32(0) == 0x00505A4D and
        any of ($x1) and
        for any j in (0..pe.number_of_sections - 1): (pe.sections[j].name == ".pdata") and
        filesize > 1MB and filesize < 3MB and
        ((pe.linker_version.major == 0x53) and (pe.linker_version.minor == 0x52))
}
```

12.2 REGLAS DE SNORT

Las siguientes reglas de Snort detectan la resolución de los dominios involucrados con Azorult; tanto la propia descarga del binario dañino como del C2.

```
alert udp $HOME_NET any -> $EXTERNAL_NET 53 (msg:"DNS request protonvpn.store (C2 Azorult malware packed with Armadillo)"; flow:to_server; byte_test:1,!&,0xF8,2; content:"|09|protonvpn|05|store|00|"; fast_pattern:only; metadata:service dns; threshold:type limit, track by_src, count 1, seconds 60; classtype:trojan-activity; sid:1000001; rev:1;)

alert udp $HOME_NET any -> $EXTERNAL_NET 53 (msg:"DNS request account.protonvpn.store (C2 Azorult malware packed with Armadillo)"; flow:to_server; byte_test:1,!&,0xF8,2; content:"|07|account|09|protonvpn|05|store|00|"; fast_pattern:only; metadata:service dns; threshold:type limit, track by_src, count 1, seconds 60; classtype:trojan-activity; sid:1000002; rev:1;)
```

13. REFERENCIAS

[REF.- 1] Threat Actors Using Legitimate PayPal Accounts To Distribute Chthonic Trojan:

https://www.proofpoint.com/us/threat-insight/post/threat-actors-using-legitimate-paypal-accounts-to-distribute-chthonic-banking-trojan

[REF.- 2] Reversing Credential and Payment Card Information Stealer 'AZORult V2':

https://www.vkremez.com/2017/07/lets-learn-reversing-credential-and.html

[REF.- 3] New version of AZORult stealer improves loading features, spreads alongside ransomware in new campaign:

https://www.proofpoint.com/us/threat-insight/post/new-version-azorult-stealer-improves-loading-features-spreads-alongside

[REF.- 4] AZORult Trojan Serving Aurora Ransomware by MalActor Oktropys:

https://www.bleepingcomputer.com/news/security/azorult-trojan-serving-aurora-ransomware-by-malactor-oktropys/

[REF.- 5] The Emergence of the New Azorult 3.3:

https://research.checkpoint.com/2018/the-emergence-of-the-new-azorult-3-3/

[REF.- 6] Twitter: Leak Builder / C2 Panel:

https://twitter.com/_CPResearch_/status/1041687670567694336

[REF.- 7] The 'Gazorp' Dark Web Azorult Builder:

https://research.checkpoint.com/2018/the-gazorp-dark-web-azorult-builder/

[REF.- 8] AZORult++: Rewriting history:

https://securelist.com/azorult-analysis-history/89922/

[REF.- 9] Seamless Campaign Delivers Ramnit via RIG EK at 188.225.82.158:

https://malwarebreakdown.com/2017/11/12/seamless-campaign-delivers-ramnit-via-rig-ek-at-188-225-82-158-follow-up-malware-is-azorult-stealer/

[REF.- 10] AZORult brings friends to the party:

https://blog.talosintelligence.com/2020/04/azorult-brings-friends-to-party.html

[REF.- 11] AZORult spreads as a fake ProtonVPN installer:

https://securelist.com/azorult-spreads-as-a-fake-protonvpn-installer/96261/

[REF.- 12] Fake Pirate Chick VPN Pushed AZORult Info Stealing Trojan:

https://www.bleepingcomputer.com/news/security/fake-pirate-chick-vpn-pushed-azorult-info-stealing-trojan/

[REF.- 13] Técnicas Mitre ligadas con AZORult:

https://attack.mitre.org/software/S0344/

[REF.- 14] RAMBO: Run-time packer Analysis with Multiple Branch Observation

https://talos-intelligence-site.s3.amazonaws.com/production/document_files/files/000/000/037/original/2016-dimva-ugarte-rambo.pdf

[REF.- 15] Armadillo 4.20 Removing the armour: a naked animal

http://www.reteam.org/papers/e72.pdf

[REF.- 16] Messing with Azorult Part 1: Malware Breakdown

https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/messing-with-azorult-part-1-malware-breakdown/