وزارة الاتصالات والتقانة

الهيئة الوطنية لخدمات الشبكة

مركز أمن المعلومات

دراسة بحثية عن البرنامج الخبيث

The Slingshot APT

إعداد

ماجد اسماعيل

رئيس دائرة الدراسات والأبحاث في مركز أمن المعلومات

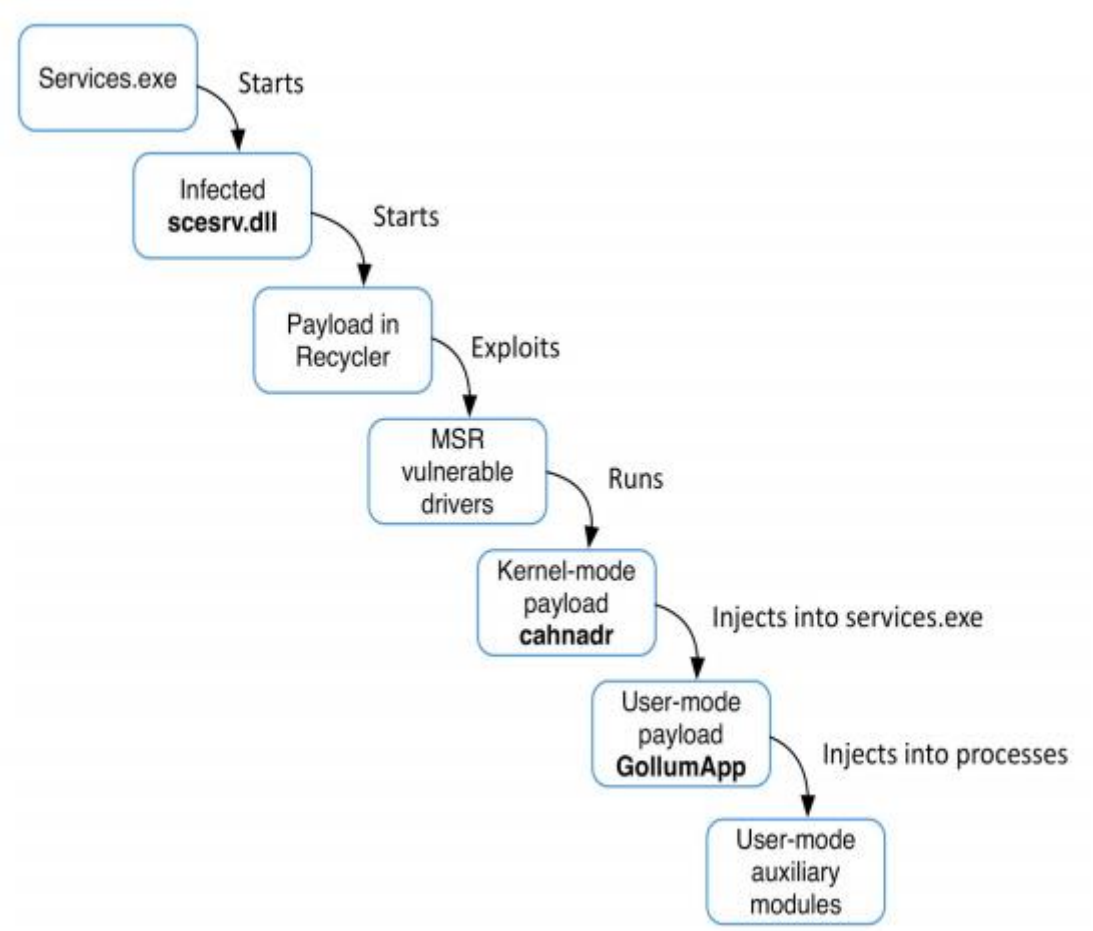
نيسان-2018

The Slingshot APT

أحد البرمجيات الخبيثة الحديثة نسبياً حيث تم تأكيد أول إصابة به في العام 2012 وهو لا يزال يعمل بنشاط حتى الآن، ينتمي لمجموعة Cyber–Espionage أو برمجيات التجسس عبر الفضاء السيبراني، يقوم باستهداف الشبكات والمنظومات الشبكية المؤسساتية، يتمتع هذا البرنامج بميزة APT: Advanced Persistent Threat أي أنه برنامج متقدم ذو طبيعة استمرارية في عمله من خلال برمجيات خارجية Command and Control تقوم بعمليات إدارة وتحكم واستخلاص البيانات من الشبكات والمنظومات المستهدفة والتي قد تمت اصابتها. والدول التي تم تسجيل أعداد متفاوتة من الإصابات ضمن منظوماتها: العراق، اليمن، ليبيا، الأردن، الصومال، السودان، الإمارات المتحدة، كينيا، أفغانستان، تنزانيا، موريشيوس، الكونغو الديمقراطية وتركيا.

المعلومات التقنية Technical Details

إن Slingshot عبارة عن برنامج متكامل حيث يستخدم العديد من الإجراءات والمكونات والمراحل:



يعتمد البرنامج بشكل رئيسي على استغلال ثغرة أو عدة ثغرات ضمن أنظمة تشغيل الموجهات Routers التي تنتجها شركة MikroTik، ومن ثم الانتشار ضمن الأجهزة المتصلة بها، حيث يعتمد الهجوم على ملفات مكتبة الربط الحيوي DLL. وبشكل رئيسي على الملف scesrv.dll، في الواقع يستبدل البرنامج هذا الملف وهو من ضمن ملفات نظام التشغيل بملف آخر يحوي رمازات ومكونات إضافية مع الحرص على أن يكون الملف الجديد بنفس حجم الملف الأساسي، عملية الاستبدال هذه تكون على عدة مراحل:

1. حقن مكونات البرنامج ضمن الملف الأساسي وهو من النوع DLL.
2. ضغط محتويات الملف الأساسي مع المكونات التي تمت إضافتها وذلك حفاظاً على حجم الملف.
3. تعديل قيم التوابع العائدة لنقطة التأشير Entry Point بحيث تشير إلى أحد المكونات المضافة.
4. إعادة احتساب القيمة Checksum للملف.
5. عند إقلاع النظام يقوم البرنامج الخبيث بتنفيذ عمل المكونات التي قد أضافها ضمن الملف DLL. ثم بعدها يقوم بإعادة تحميل الرماز الأصلي للملف الأساسي ضمن ذاكرة النظام.

المكونات الأساسية Main Modules

Ring0 Loader

العنصر المسؤول عن تحميل الوحدة الرئيسية للبرنامج، وقد قام مطورو البرنامج Slingshot بإنشاء أكثر من نسخة من هذا العنصر حرصاً منهم على إعادة تشغيل نسخة ثانية منه في حال حدوث أي فشل محتمل أثناء تشغيل النسخة الأولى حيث سيقوم بعدها بالمهام التالية:

- اكتساب السماحيات SeLoadDriverPrivilege للقيام بتنصيب الملفات الضرورية وذلك من أجل الحصول لاحقاً على السماحيات Kernel Privileges للعمل ضمن نواة النظام.
- تجاوز عمليات أرشفة الأحداث ضمن نظام التشغيل حيث يقوم بإعادة تسمية السجلات ETW-logs
- إضافة اللاحقة tmp. حرصاً على التعمية على أسماء الملفات المؤقتة التي يقوم بإنشائها أثناء عمله، علماً بأنه سيقوم بحذفها لاحقاً.
- المهمة الرئيسية والنهائية هي تحميل الوحدة Cahnadr ضمن بيئة عمل النواة Kernel Mode.

Cahnadr: Main Kernel Mode Payload

المكون الأساسي للبرنامج، يعمل ضمن نواة النظام وهو المسؤول عن إدارة وتنظيم عمل وحدات البرنامج الأخرى، يقوم بالعديد من المهام منها:

- Anti-Debugging Actions مكافحة عمليات تصحيح نواة النظام والتحقق المستمر من أن نواة النظام قد تم تصحيحها أم لا Patched or Not ويتم ذلك من خلال العديد من التقنيات أهمها إعادة احتساب القيمة CheckSum بالإضافة إلى مراقبة الطابع الزمني TimeStamp العائدة للنواة، وذلك لأن إعادة تصحيح أو ترقيع نواة النظام قد يتسبب بإعاقة عمل البرنامج.
- استدعاء خدمات النظام System Services بشكل مباشر Calling Directly وذلك من أجل إخفاء الأنشطة العائدة للبرنامج الخبيث.
- مراقبة النسخة الحالية لجدول الخدمات العاملة ضمن نواة النظام KTHREAD.ServiceTable
- بما أن هذا المكون يعمل ضمن نواة النظام فهو بالطبع يقوم بمهام الجذور Rootkits وخاصة تلك المهام المتعلقة بإخفاء بيانات الدفق الشبكي المتعلقة بعمل البرنامج كالبيانات التي يقوم بإرسالها.
- حقن حمولة الوحدة العاملة ضمن بيئة المستخدم User-Mode Payload ضمن ملف الإجراء services.exe
- تجهيز واجهات برمجة تطبيق خبيثة Malicious APIs اللازمة لعمل مكونات البرنامج العاملة ضمن بيئة المستخدم User Mode Modules
- تجهيز كل ما يلزم من أجل عمليات الاتصال Network Communications بواسطة الاتصالات الشبكية المختلفة.
- تنبيه وتزويد الوحدة GollumApp وهي الوحدة الرئيسية العاملة ضمن بيئة المستخدم User Mode تزويدها بالأحداث Events ذات الصلة بالإجراءات العاملة بالإضافة إلى تأمين الواجهات Interfaces اللازمة لإدارة المساحات المحجوزة لهذه الأحداث في الذاكرة.
- مراقبة كافة التجهيزات (البطاقات) الشبكية Network Devices العاملة ضمن نظام التشغيل.
- التجسس على كل من البروتوكولات: ARP, TCP, UDP, DNS, ICMP, HTTP

كما تقوم الوحدة الرئيسية للبرنامج Cahnadr بكل من المهام التالية التي تتعلق ببيئة الواجهات البرمجية API:

- الوصول المباشر إلى الأقراص الصلبة قراءةً وكتابةً.
- إجراء عمليات القراءة والكتابة ضمن الذاكرة.

- حقن رمازات تنفيذية ضمن الإجراءات العاملة وتشغيلها كإجراءات فرعية Threads
- الحصول على شهادات الوصول Access Tokens بواسطة معرف الإجراء Process_ID
- الحصول على عنوان جدول تفاصيل مؤشرات الخدمات SERVICE_DESCRIPTOR_TABLE
- الحصول على مؤشر ملفات التشغيل DRIVER_OBJECT Pointer باستخدام اسماء هذه الملفات.
- جمع معلومات تفصيلية عن الإجراءات التي تعمل ضمن الملف csrss.exe
- اعتماد منهجية بدء إجراء عن طريق إجراء ثاني، حيث يكون الإجراء الثاني هو المتحكم بالإجراء الأول.
- إغلاق المقبض Handle العائد لأي إجراء.
- القيام بمهام متعددة ذات طبيعة شبكية Network Related Tasks، بالإضافة إلى جمع معلومات حول المهام الشبكية الفعالة وإرسالها إلى الوحدة GollumApp
- التعامل Hooks مع إجراء محدد ويشمل ذلك إعداد وحذف خطاف Hook بواسطة معرف الإجراء الفرعي ThreadID أو إعداد وحذف خطاف Hook لكافة الإجراءات الفرعية بواسطة معرف الإجراء PID ويتم كل ذلك بواسطة جدول الخدمات العاملة في KTHREAD
- تحديد زمن الإسبات Time to Sleep قبل إيقاف التشغيل.

بالنسبة للتعامل مع الدفق الشبكي تقوم الوحدة Cahnadr ومن أجل إخفاء الدفق الشبكي الخاص بالبرنامج الخبيث وإجراء العديد من المهام المختلفة بالإضافة إلى توفير عدد من المهام الأخرى لمكونات البرنامج العاملة ضمن بيئة المستخدم User Mode، من أجل كل ذلك تقوم الوحدة بعمل Hook لكل من الإجرائيات التالية:

1. ndis!NdisMSendNetBufferListsComplete
2. ndis!NdisMIndicateReceiveNetBufferLists

يعد التجسس Sniffing على الدفق الشبكي من أهم ما تقوم به الوحدة Cahnadr ومن ضمن عمليات التجسس هذه يتم تجهيز العديد من المهام الفرعية الأخرى Tasks من أجل البحث والتنقيب ضمن رزم الشبكة الصادرة والواردة Inbound and Outbound Network Packets ثم إخفاء تلك الرزم التي يثبت أنها تتبع للبرنامج الخبيث، ومن هذه المهام:

HTTP Task

من أجل تنبيه وإخطار الوحدة GollumApp User Mode Module بوجود عملية نقل سواءً إرسال أو استقبال بيانات بواسطة البروتوكول HTTP.

ARPf Task

تنبيه وإخطار الوحدة GollumApp في حال ورود طلب بالبروتوكول ARP وفي حال إرسال استجابة ARP، بالإضافة إلى جمع كل هذه المعلومات وتخزينها على شكل بيانات فيزيائية.

IP2f Task

ويقع على عاتق هذه المهمة تحديد فيما إذا كانت الرزم الشبكية تتبع لإحدى وحدات ومكونات البرنامج أم لا لغاية إخفائها Hidden، تتم مرحلة تمييز الرزم بواسطة عملية XORing بين قيمتين زمنيتين Timestamps Values يتم الحصول عليهما من الحقل Options من الترويسة TCP Header ومقارنة النتيجة بالقيمة الثابتة 0xDEADFOOD وفي حال المساواة فإن الرزمة الحالية تتبع للبرنامج ويتم إخفاؤها وإخطار الوحدة GollumApp بوجود رزم شبكية TCP/UDP or ICMP مطابقة لعمل البرنامج أو تتعلق بأحد مكوناته.

وظائف إضافية تؤمنها الوحدة Cahnadr من أجل عمل الوحدات العاملة ضمن بيئة المستخدم:

- ARP-query: الحصول على العنوان الفيزيائي MAC العائد لعنوان رقمي IP ويتم ذلك بواسطة بطاقة شبكية فعالة تعمل ضمن شبكة.
- ARP-replay: إرسال العنوان الفيزيائي الخاص بها (بالوحدة Cahnadr) كاستجابة لطلب ARP-request وذلك بغض النظر فيما إذا كان العنوان الرقمي IP الذي قام بهذا الطلب يعود للحاسب المصاب بالبرنامج الخبيث أم لا.
- إرسال رزم شبكية مخصصة حيث تكون كافة حقول هذه الرزم قابلة للتخصيص من خلال طبقة الشبكة Ethernet Layer والتي تعادل في المعيار OSI الطبقتين الأولى والثانية.
- إرسال رزم شبكية IPv4

من الجدير بالذكر وكميزة أخيرة للوحدة Cahnadr هو أنها تدعم العمل بالمعايير IEEE 802.11، وهذا بالطبع يمكنها من التعامل مع رزم وإطارات الشبكات WiFi. يتم عادةً تعقب عمل البطاقات الشبكية بواسطة مجموعة سجلات الأحداث EventCategory:

PNPNOTIFY_DEVICE_INTERFACE_INCLUDE_EXISTING_INTERFACES

تقوم الوحدة Cahnadr باستطلاع محتويات سجلات الأحداث هذه والخاصة بالبطاقات الشبكية من أجل تعقب أي تغييرات تتعلق بهذه البطاقات وخاصة تلك التي تتعلق بإضافة بطاقة شبكية جديدة. بالإضافة إلى ذلك تجمع الوحدة Cahnadr بيانات مختلفة منها:

- Ethernet: العنوان الفيزيائي MAC والطول الأقصى للإطار Maximum Frame Size
- Wireless: العنوان الفيزيائي MAC ومعلومات عن عملية المصادقة Authentication State

GollumApp: User Mode Payloads

الوحدة المسؤولة عن الأنشطة المتعلقة بالعمل ضمن بيئة المستخدم، بالإضافة إلى مسؤوليتها عن إدارة وتنظيم عمل الوحدات الأخرى ضمن بيئة المستخدم User Mode، وبالطبع التنسيق والتفاعل المستمر مع الوحدة الأساسية Cahnadr، وهي تبدأ العمل مباشرة بعد حقنها ضمن ملف الإجراء services.exe من قبل الوحدة Cahnadr وكأي إجراء آخر تقوم بحجز مساحة معينة ضمن الذاكرة ثم كتابة الوحدات التنفيذية ثم إنشاء إجراء فرعي جديد Thread، ويمكننا اختصار مهام ومراحل عمل الوحدة GollumApp بما يلي:

- جمع المعلومات المتعلقة بالأداء الشبكي للنظام مثل: جداول التوجيه Routing Tables، الإعدادات الشبكية Network Configurations، معلومات عن المخدمات الوكيلة Proxy Servers بالإضافة إلى المعلومات المتعلقة بكيفية إدارة وتنسيق الروابط AutoConfigUrl.
- جمع الإشعارات Notifications المتعلقة بكافة التعديلات التي قد تطرأ على جداول التوجيه Routing Tables بالإضافة إلى الإشعارات المتعلقة بكافة التعديلات على العناوين الرقمية العائدة للبطاقات الشبكية Interfaces IPs.
- التعامل Hooks مع طلبات الإدخال والإخراج IO Requests الخاصة بملفات نظام التشغيل المشفرة.
- تحوي العديد من الأوامر والتعليمات المخصصة للتعامل مع أنظمة CNC.
- جمع كافة كلمات المرور التي تم حفظها ضمن كل من المتصفحات Mozilla, Internet Explorer
- التعامل مع حافظات النسخ Clipboard في نظام التشغيل.
- العمل كمسجل للمفاتيح Key logger ويشمل ذلك كافة المفاتيح التي يتم الضغط عليها.
- جمع معلومات عن أقسام الأقراص الصلبة HDD Partitions
- جمع معلومات عن التجهيزات التي تتمتع بواجهة توصيل نوع USB مع التحسس لأي تجهيزات جديدة من هذه النوع يتم وصلها بالنظام وتسجيل وإرسال إشعارات بعملية التوصيل.

- يستطيع تشغيل إجراء جديد بسماحية النظام SYSTEM Privileges كإجراء ابن للإجراء smss.exe
- حقن الوحدة SsCb وهي وحدة فرعية تتبع للوحدة GollumApp وذلك ضمن إجراء معين Into Specified Process.

وحدات أخرى تعمل ضمن بيئة المستخدم Auxiliary User Mode Payloads

SsCb

توفر هذه الوحدة كل من المهام التالية:

- إجراء لقطات شاشة Screenshots لسطح المكتب أو لإطار معين.
- الحصول على محتويات حافظة النسخ Clipboard
- جمع معلومات عن النوافذ المفتوحة كالعنوان، الحجم والموضع.
- إغلاق أي نافذة مفتوحة بواسطة إرسال الرسالة WM_CLOSE
- إظهار أو تفعيل أي نافذة بواسطة استدعاء الأمر ShowWindow
- جمع معلومات مختلفة عن سطح المكتب الفعال، النافذة الفعالة، اسم الإجراء الذي قام باستدعاء النافذة، عنوان النافذة والتخطيط العام للوحة المفاتيح.

ffproxy

تجمع هذه الوحدة معلومات تفصيلية عن إعدادات المخدمات الوكيلة Proxy Settings وذلك للمتصفحات التابعة لشركة Mozilla: المخدمات الوكيلة لكل من البروتوكولات HTTP, SSL، معلومات عن autoconfig-url وهو ما يتعلق بالتحديد الآلي للمخدمات الوكيلة وبما يشمل المحلي والبعيد Local or Remote بالإضافة إلى أسماء النطاقات وأرقام المنافذ وحتى الحسابات التي يتم الدخول عن طريقها إلى المخدمات الوكيلة حيث تقوم هذه الوحدة بالحصول على كل من أسماء المستخدمين وكلمات المرور.

NeedleWatch

يتم حقن هذه الوحدة الفرعية ضمن معظم الإجراءات التي يتعلق عملها بكل من الوحدتين GollumApp, Cahnadr، حيث إن عملها الرئيسي هو التجسس على محتويات الذاكرة الوسيطة Buffers والتي يتم تمريرها لكل من التوابع التالية:

- التوابع التي تستخدم في عملية إظهار النصوص Draw Text:
  - gdi32!ExtTextOutW
  - gdi32!ExtTextOutA
  - gdi32!TextOutA
  - gdi32!TextOutW
- التوابع التي تستخدم في الكتابة ضمن الـ Console
  - kernel32!WriteConsoleA
  - kernel32!WriteConsoleW
- التابع المستخدم في إظهار ترميز النص Rendering Unicode Text:
  - usp10!ScriptShape
- التابع المستخدم في إظهار النص بواسطة الإجراء DirectWrite:
  - dwrite!DWriteFontFace::GetGlyphIndicesW
- التوابع المستخدمة في عمليات التشفير وفك التشفير ضمن وحدة Security Support Provider:
  - secur32!EncryptMessage
  - secur32!DecryptMessage
- التوابع المستخدمة ضمن بيئة التشغيل Netscape Portable Runtime:
  - nspr4!PR_GetUniqueIdentity
  - nspr4!PR_Read
  - nspr4!PR_Write

Sfc2

تقوم هذه الوحدة بعملية إيقاف وتعطيل منهجية حماية الملفات SFC: System File Checker.

آلية استغلال الثغرة لإصابة الموجهات Infecting MikroTik Routers

شركة MikroTik هي إحدى الشركات المصنعة للتجهيزات الشبكية، وقد قامت الشركة بتزويد مالكي الموجهات Routers بتطبيق WinBox من أجل إدارة عمل هذه الموجهات، يعتمد التطبيق على تحميل عدد من الملفات ذات النوع DLL. مباشرة من نظام ملفات الموجه ورفعها إلى ذاكرة الحاسب، وهذه هي الطريقة الطبيعية المتبعة

لإدارة الموجهات التي تنتجها شركة MikroTik؛ يعتمد المهاجم في استغلاله لهذه المنهجية على رفع ملف ip4.dll إلى نظام ملفات الموجه حيث سيقوم التطبيق WinBox بتحميله من الموجه ثم رفعه إلى ذاكرة الحاسب ضمن المسار %AppData%\Roaming\mikrotik\"Firmware Version Name"\ipv4.dll ، هذا الملف يتميز بالاسم الداخلي chmhlpr.dll وهو عبارة عن برنامج يقوم بتحميل برمجيات خبيثة أي من النوع Trojan Downloader وهو يتبع للبرنامج الخبيث Slingshot. بحسب الشركة المصنعة فإن هذه الثغرة موجودة فقط في نسخ أنظمة تشغيل الموجهات حتى الإصدار RouterOS v.6.38.4، وأن النسخ المحدّثة من التطبيق WinBox لم تعد تقوم بتحميل الملف ip4.dll من نظام ملفات الموجه.

chmhlpr.dll

كما تم ذكره سابقاً هذا هو الاسم الداخلي للملف الخبيث ip4.dll وهو من النوع Trojan Downloader أي أن عمله هو تحميل المزيد من الملفات الخبيثة حيث يقوم هذا الملف/البرنامج بتحميل حزمة ملفات MZPE وهي ملفات تنفيذية مزودة بكل من التفاصيل التالية:

- عنوان رقمي IP من أجل تحميل الحمولة Payload وقد تم إيجاد هذه الحمولة ضمن الموجهات المصابة ذات العنوان الرقمي 192.168.88.1
- رقم المنفذ الذي سيتم الاتصال عبره، وقد تم تحديد المنفذ رقم 4443
- عدد محاولات إجراء وإعادة الاتصال، وقد تم تحديد عدد 3 محاولات.
- المدة الزمنية الفاصلة بين محاولات الاتصال، وقد تم تحديد 90 ثانية.

علما بأن هذا الـ Trojan Downloader يستخدم معلومات المخدمات الوكيلة من المسار التالي:

UserSID\Software\Microsoft\Windows\CurrentVersion\InternetSettings\ProxyServer*

مع إمكانية البحث عن بيانات الدخول إلى حساب Proxy في:

- ضمن منطقة التخزين Windows Protected Storage حيث يحوي المتغير ItemName اسم مجال المخدم الوكيل الحالي Proxy Domain
- بيانات الدخول المحفوظة ضمن التصفح IE

KPWS

ملف تحميل Downloader آخر ولكنه مختلف عن الملف chmhlpr.dll وهو يقوم بتحميل مكونات أخرى تتبع للبرنامج Slingshot وقد ثبت بأن هذه الأداة تحوي عدداً من المؤشرات التي تشير إلى التطبيق GollumApp

ملفات تحميل أخرى Additional Downloaders

Rc

ملف أو أداة تحميل تشبه إلى حد بعيد الملف chmhlpr.dll فيما يخص متغيرات الدخل Input Parameters وبيانات الخرج Output askpws وهي تقوم بما يلي:

- حل قيم متغيرات البيئة Resolve environment variables
- إرسال معلومات حول الملفات ضمن الدليل كالمسار والحجم وتاريخ آخر تعديل.
- إرسال معلومات حول الإجراءات العاملة PID, PPID, ، زمن بداية التشغيل، اسم الملف التنفيذي الذي يشغل الإجراء، اسم الحساب مع اسم المجال ومعلومات أخرى.
- إيقاف عمل إجراء باستخدام الرقم المعرف PID
- انتحال شخصية المستخدم ويتم ذلك بواسطة الاسم وكلمة المرور بعد الحصول على هذه البيانات من المخدم أو عن طريق رقم المعرف PID
- إعادة وضع العمل إلى الوضع العادي بعد عملية انتحال المستخدم.
- إنشاء إجراءات جديدة بالوضع العادي أو بعد انتحال المستخدم.
- التواصل مع الإجراءات التي قام بإنشائها.
- إرسال كل من المعلومات: اسم الحاسب الحالي، إصدار نظام التشغيل Windows version ، رقم بناء النسخة Build Number بالإضافة إلى رقم تحديث Service Pack الحالية.
- إرسال اسم المستخدم.
- الهجرة Migrate إلى العمل ضمن إجراء آخر حيث تقوم الأداة بإصابة إجراء آخر بواسطة رقم المعرف PID وتتم الإصابة ضمن الذاكرة مع تمرير مقبس الاتصال Socket بالمخدم أيضاً.
- الهجرة Migrate إلى العمل ضمن إجراء آخر حيث تقوم الأداة بإنشاء إجراء بعد استقبال مساره من المخدم عندها تحقن الأداة نفسها ضمن الإجراء الجديد في الذاكرة فقط مع تمرير مقبس الاتصال مع المخدم عندها تقوم الأداة بتحميل وتنفيذ الموحدة التالية من البرنامج Slingshot
- تحميل وإعداد وحدات جديدة New Modules وتشغيل هذه الوحدات في إجراء فرعي جديد New Thread ضمن الإجراء الحالي Current Process مع إرسال بيانات دخول الوحدات الجديدة إلى المخدم المتصل.

Spork Downloader

يختلف هذا الملف عن ملفات التحميل التي تم عرضها سابقاً وذلك قياساً بالمهام الإضافية المكلّف بها حيث يعتبر هذا الملف/البرنامج كمحرك للقواعد Rule Engine وهو مزود بالعديد من القواعد الجاهزة Serialized Rules، والغاية الرئيسية من هذه البنية هي البحث عن أدوات الحماية الشخصية Personal Security Products PSP ضمن النظام المستهدف كالبرامج المضادة للبرمجيات الخبيثة، وتحديد القواعد المناسبة للعمل استناداً إلى الإجراءات الفعالة Started Processes بحيث يتم تحديد الإجراء الفعال المناسب من أجل حقن الرماز الخبيث ضمنه. وقد تم تصنيف (تسلسل) القواعد بحسب كل من المخططات Scheme التالية:

- Byte count_rules, count_PSPs
- Rule all_rules [count_rules] (6 or 8 bytes per rule)
- Short offsets_to_PSP_names [count_PSPs]
- Char PSP_names [count_PSPs]

تتألف كل قاعدة Rule من الحقول التالية:

- اسم الإجراء العائد لأداة الحماية PSP وهو مصنف كفهرس ضمن مصفوفة.
- مصفوفة أسماء الإجراءات التي سيتم الحقن ضمنها وهي مفهرسة أيضاً.
- الإصدار الأدنى Min من نسخة أداة الحماية.
- الإصدار الأعلى Max من نسخة أداة الحماية.
- بعض المؤشرات الأخرى Flags مثل بيئة العمل x32, x64
- نمط Type يستخدم كنتيجة مرجعية عند التثبت من وجود القاعدة.

تقتضي مهمة البرنامج الفرعي Spork في أنه يستطلع كافة الإجراءات الفعالة ويطابق كل منهم مع كل قاعدة، وفي حال مطابقة أحد هذه الإجراءات مع قاعدة واحدة على الأقل عندها يتم تحديد ما إذا أنه سيتم الحقن في هذا الإجراء أم لا وذلك بحسب نمط Type القاعدة المطابقة للإجراء، وقد تكون قيمة هذا النمط واحدة مما يلي:

- Type 0: الافتراضية.
- Type 1: خطأ.
- Type 2: حقن ضمن PSP المطابقة.
- Type 3: حقن ضمن الإجراء lsass.exe
- Type 4: حقن ضمن الإجراء winlogon.exe

- Type 5: حقن ضمن الإجراء svchost.exe
- Type 6: حقن ضمن الإجراء المحدد في الحقل التالي من القاعدة المطابقة.

وفي حال عدم مطابقة أي من الإجراءات لأي قاعدة فإن عملية حقن الرماز ستتم ضمن الإجراء svchost.exe وذلك بشكل افتراضي. يمكن تلخيص محددات مطابقة الإجراء لقاعدة ما بما يلي:

1. اسم الإجراء مطابق لاسم أداة PSP الوارد ضمن القاعدة.
2. إصدار أداة PSP متضمن في المجال الوارد ضمن القاعدة أي بين الحقلين Min, Max
3. يجب أن تكون المؤشرات Flags الواردة ضمن حقول القاعدة متناسبة مع الإجراء.

تتم عملية تحديد اسم وإصدار أداة الحماية PSP بواسطة عدد من الاستدعاءات لكل من GetFileVersionInfo و VerQueryValue من أجل الحصول على بيانات الحقل dwProductVersionMS والذي يحوي المعلومات المطلوبة حول أداة الحماية. يبين الجدول التالي أسماء أدوات الحماية PSP بالإضافة إلى اسم الإجراء المطلوب للحقن، وذلك بشكل مختصر علماً بأن هذه البيانات قد تم تحديدها من خلال الأبحاث العملية:

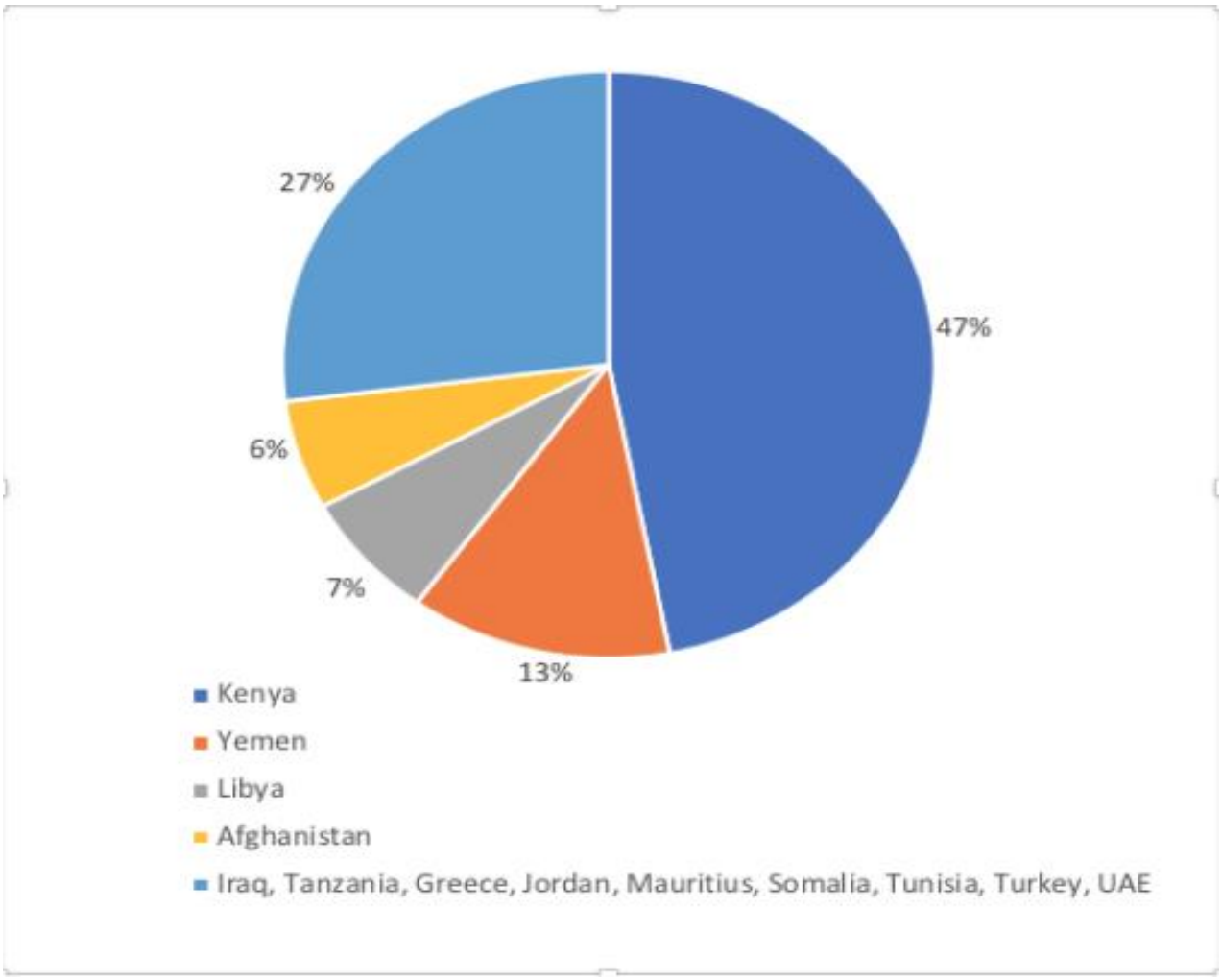
Found PSP name	Versions	Bitness	Process to inject
avfwsvc.exe	00-ff	x32	avguard.exe
avfwsvc.exe	00-ff	x64	inssda64.exe
avgtray.exe	00-ff	x32	avgtray.exe
avgtray.exe	00-ff	x64	avgsrmaa.exe
avp.exe	01-07	x32-x64	winlogon.exe
avp.exe	08-0c	x32	avp.exe
avp.exe	08-0c	x64	lsass.exe
avp.exe	0d-0d	x32-x64	lsass.exe
avastui.exe	00-ff	x32	avastui.exe
avastui.exe	00-ff	x64	winlogon.exe
avgnt.exe	00-ff	x32	avguard.exe
avgnt.exe	00-ff	x64	inssda64.exe/avshadow.exe

avgui.exe	00-ff	x32-x64	winlogon.exe
bdagent.exe	00-ff	x32-x64	bdagent.exe
cfp.exe	00-ff	x32-x64	cfp.exe
casc.exe	07-08	x32-x64	svchost.exe
casc.exe	05-06	x32-x64	Error
defenderdaemon.exe	00-ff	x32-x64	Error
egui.exe	00-ff	x32-x64	default - svchost.exe
fsdfwd.exe	00-ff	x32-x64	default - svchost.exe
mcagent.exe	00-ff	x32-x64	winlogon.exe
rstray.exe	00-ff	x32	rstray.exe
rstray.exe	00-ff	x64	Error
rtvscan.exe	00-ff	x32-x64	default - svchost.exe
tmproxy.exe	00-ff	x32-x64	tmproxy.exe
umxcfg.exe	07-08	x32-x64	default - svchost.exe
umxcfg.exe	05-06	x32-x64	Error
zlclient.exe	00-ff	x32-x64	Error

لا تعتمد الأداة Spork على الحقن المباشر ضمن الإجراءات الفعالة، بل تقوم بإنشاء إجراء جديد لنفس الملف التنفيذي Image File، يتم إنشاء الإجراء الجديد بالصيغ x32, x64 وبالمحددات التالية: flags hide, create no window, default instead of loading cursor and suspended، ومن ثم إنشاء قطاع جديد Section وإدخال الرماز الخبيث المطلوب في هذا القطاع، وذلك اعتماداً على الإجراء الذي تم إنشاؤه ثم تعديل نقطة التأشير Entry Point بحيث تقوم باستدعاء الرماز الخبيث، وأخيراً تقوم الأداة Spork باستدعاء ResumeThread وذلك من أجل تشغيل الإجراء.

المنظومات التي تمت إصابتها Victims

تتركز معظم الإصابات في منطقة الشرق الأوسط وافريقيا وفيما يلي مخطط يوضح مكان تواجد المنظومات والشبكات التي تمت إصابتها مع النسب المئوية محددة بحسب العدد التقريبي للإصابات وذلك حتى شباط 2018:



مصادر الدراسة والصور والإحصائيات:

- دراسات وأبحاث الشركة الروسية Kaspersky Lab.