eCTHP V2

Malware Hunting

BY: ***Ahmad Abdelnasser Soliman***

abdelnassersoliman0@gmail.com

Index of Content:

1.Introduction to malware hunting:

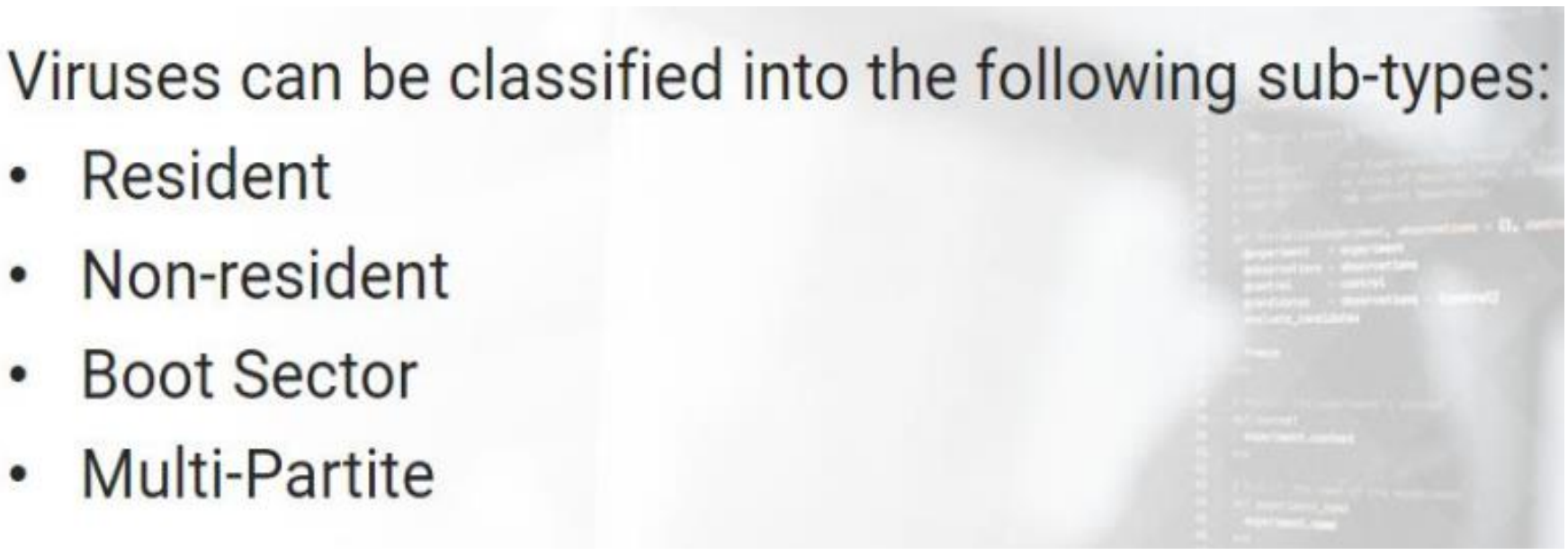
- ال malware اختصارا ل Malicious Software فهي برمجيه خبيثه على شكل كود برمجي وعموما اشكاله كتير لما يجي يعمل Infection لل End point target فيه ال DLL وال exe وال Ps ال هو ال Power Shell ودي معظم اشكاله ال بتلاقيها موجوده عندك.

- هدفه انه يعمل Damage أو Filtrate يعني يسرب Data أو يخرب ال PC System عند ال End point بدون مصاحب ال Machine دا يكون عارف حاجه فتلاقيه شغال فال Background عند ال Target وانت عادي بس تتفاجيء بالمشكله قدامك زي ايقونه برنامج معين تظهرلك فجأه أو ملفات جهازك تتشفر مقابل مبلغ مادي وهكذا.

2.Malware classification:

- أول حاجه هنتكلم عنها ف انواع ال Malware هي ال Virus ودا Malicious Software بيعمل Copy لنفسه نتيجه لان ال User ال مترجت فال Attack دا عمل Action معين على جهازه خلى ال Virus دا يشتغل عال End point ... مثلا عند ملف عال System عندك اسمه firefox.exe ملف تنفيذي عشان تثبت المتصفح ولكن الملف دا انت منزله من موقع مش مضمون فالملف مش Clean فالملف دا لازق فيه ال Virus ولكن مش هيشتغل الا اما تتفاعل معاه انت ك User بمعني تعمل للملف دا تثبيت أو نقل من مكان لأخر أو copy المهم لازم تدخل منك ودا الفرق بين ال Virus وال Worm ال Worm مش بتحتاج من ال User اي تدخل مجرد متنزل على ال End point هي بتنتشر لوحدها وتنفذ ال Infection لل target بتروح تدور على Vulnerability تعملها Exploit ...

وترجتها عشان تشتغل من خلالها انما ال Virus بيحتاج تدخل منك انت ك User لل File ال Infected .... تعالى نشوف تصنيفات وانواع ال Virus .



- ال Resident يعني مقيم عندك فال End point بمعني ال Files ال Infected الموجود فيه ال Virus لو عملتله execute هتلاقيه بيشتغل عادي وبرضه متواجد عندك فال Memory الخاصه بال End point مبيخرجش من الجهاز بمعنى اي حاجه عاوز تشغلها عندك فال End point زي ملف او لعبه او فيلم بتروح تاخد مكان فال Memory الاول بيتحجزلها مكان وبعدين تشتغل أو بيتاخد منها نسخه هناك ...

- فتخيل انت شغلت ملف infected ب virus هتلاقيه راح اشتغل جوا ال RAM وفيه نسخه من ال virus قاعده جوا ال RAM مستنياك بس تشغل اي exe file أخر برضه هيروح ال RAM عشان يشتغل نفس القصه هنا ال Virus يبدء يعمله Infect ومن هنا بيجي حته ال Spread لل Virus على جهازك .

- النوع التاني من ال Virus معانا وهو ال Non-Resident ودا بيدور عاى Files معينه جوا ال System عندك عشان يعملها ال Infection وبعد اما يعملها ال Infection هيخرج من جهازك وانت ك User اما تيجي تشغل ال Files التانيه هتلاقي ال Virus ...

اشتغل معاها وبرضه هينتشر ويعمل ال infection effect بتاعه عادي لكن من الملفات ال اصابها الملف الاساسي المصاب بال Virus ولكن خرج من ال System وساب الملفات ال Infected تشتغل بداله

- النوع التالت عندنا وهو ال Boot Sector ودا بيشتغل مع ال Boot لل System بتاعك هو موجود فالمكان الخاص بأقلاع النظام فأنت تيجي تعمل Booting لجهازك يشتغل معاه علطول ... زي مثلا جيت تعمل Booting لل System وحطيت USB قبلها وال Virus موجود فمكان الاقلاع وشغلت انت ال PC و هو عمل Infected عندك لل USB ال انت وصلتها بالملفات الخاصه بيها ... فأي Device موصله بالجهاز وانت بتعمل Booting هيتصاب بال Virus .

- عندنا النوع الرابع وهو ال multi-partite ودا بيبقا كذا نوع مع بعض يعني ال Boot Sector مدموج مع ال Resident مثلا يعني Virus واحد بيقوم بدورالاتنين مع بعض وال Malicious code بتاعه مصمم لكدا ودا طبعا أخطرهم لانك ممكن توظفه يعملك كذا Task يصيب ال Files الموجوده فال Memory وال Boot Sector .

- تاني نوع عندنا وهو ال Worm ... ببساطه دي اي Software بيستخدم ال System or Network Vulnerability عشان يعمل لنفسه انتشار ... بتستغل اي ثغرات فالنظام او الشبكه عشان تعمل منها انتشار لنفسها عالنظام ال مترجت يتعمله ال Infection ... يعني احنا لو ف Network وجهاز مصاب ب Worm معينه ترجتت ثغره معينه لقتها فيه واصابتها وانت معاهم فنفس ال Network لو عندك نفس الثغره فجهازك فكدا كدا ال Worm هتجيلك وتعملك ال Infection لانها بتعمل Search على الثغره حاليا فكل الاجهزة الموجوده عال Network .

ـ طب لو انت معندكش الثغره دي أوعملها Batching أو جهازك معموله update ومقفول فيه الثغره دي فساعتها ال worm مش هتقدر تصيبك لانك قافل الباب ال بتدخل منه وهي الثغره.

ـ فيه نقطه تانيه معانا وهي ان ال Worm مبتجيش لجهازك لوحدها كدا لان من الطبيعي هيتعملها Detect لاء ... دي بتيجي معاها ال Rootkit ودا هنتكلم عليه فدوره الجي ودا نوع من انواع ال malware المسؤول عن انه يخفي اثار ال Malware ال شغال عندك على جهازك وبيخرب فيه زي ال Worm كدا ... ال Rootkit مسؤول انه يخفيه عن ال Antivirus ال انت ك User مشغله عندك وطبيعي انه يعمل Detect لل Worm فكدا كدا بيجي معاها ال Rootkit طبيعي عشان يعملها Hide بعيدا عن انظار ال User أو ال Scanner software ال هتعملها Detect .

ـ تالت نوع من ال malware هنتكلم عليه هو ال Rootkit ودا زي مقولنا مسؤول انه يعمل hide لل malware ال شغال عند ال Victim عشان ميتعملوش detect من ال Antivirus Software ال Rootkit بيساعدك تخفي ال Processes الخاصه بال malware عندك ال Victim وكمان يقدر ينشألك backdoor عند ال Victim ...

ـ بيستخدموه ال Attackers فال Maintaining Access لما يحبوا يرجعوا لجهاز ال Victim ال Infected فأي وقت ... وكمان يقدر Add ل Files معينه تخص ال Malware فال File System ويخفيها جواه بحيث ال Antivirus ميعرفش يجيبها ... وكمان يقدر يعملك create ل Loopholes بمعنى متعرفش تتبع ال malware على جهازك كل متوصل لطرف خيط انك مثلا تعرف الملفات الخاصه بال Malware دي موجوده فين يرجعك ال Rootkit لنقطه الصفر .

ـ فممكن يغير مسار الملفات الخاصه بال Malware ال انت بتدور عليه وهكذا ... ال Rootkit عنده ال Technique الخاص بيه ال يقدر يستخدمه عشان يوقعك ف حفره واحده ورا التانيه ومتوصلش لمكان ال Malware أو الملفات الخاصه بيه ودا المقصود بال Loopholes .

ـ خطورة ال Rootkit تتمثل في انها لما بتصيب ال System بيبقى ليها تحكم كامل فال Operating System الخاص بال Victim ودا لان ال Rootkit بينزل نفسه فال Layer دي وال Antivirus بيبقا شغال فال Application layer فمبيعرفش يوصل لل Layer دي ولا يشوف ال Rootkit من أساسه ... فتلاقيه بيشغل نفسه مع ال OS كاءنه File من ضمن ال Files الخاصه بال Kernel الخاصه بالنظام . فهتلاقي من ضمن الاماكن اللى موجود فيها ال Rootkit هي ال Library Level او ال Kernel Level او ال Application Level أو ال Firmware Level أو ال Hypervisor Level .

ـ عندك ال Application Level دا سهل تكتشف فيه ال Rootkit عشان بيسمى نفسه بأسماء Programs عندك عال System فدا ال Antivirus بيعمله Detect ... انما ال Library Level دا بيعمل Infected ل Library معينه زي المكتبه الخاص بالصوت وزي المكتبه الخاصه بالفيديو وهكذا كل حاجه عندك ليها مكتبه بتعملها استدعاء عشان تشغلها ...

ـ تخيل انت لو عندك 10 Applications شغالين بنفس ال Library يعني وليكن كلهم بيستخدموا ال Library الخاصه بالصوت ... وانت عندك Rootkit عمل Infected لل Library الخاصه بالصوت ... يعني عمل Infected لل 10 Applications كلهم عشان شغالين بنفس ال Library .

- عندك ال Kernel Level ودا الاكثر شيوعا لل Rootkit ودا لانه بيقدر من خلال تواجده فال Kernel لل OS انه يعمل ال Resistance لل AV لانه واخد صلاحيات ال AV وال AV ميقدرش يشيل أو يحذف الاقل منه فالصلاحيات انما ال زيه ميعرفش ...

- عشان كدا بتلاقي معظم ال Rootkit بتبقى مصممه من ناحيه ال malware Developer انها تروح تترجت ال Kernel الخاص بال OS .

- عندك ال Hypervisor Level ودا الخاص بال Processors الحديثه الخاصه بال Virtualization Programs زي ال VMware وال Virtual box ال بتمكنك انك تشغل أكتر من نظام تشغيل عندك على جهازك بيبقى عباره عن نظام افتراضي ... اهو دا فيه Rootkit خاص بيه بترجت ال Processor الخاص بال VM .

- عندك ال Firmware Level ودا بيتواجد فيه الملفات الثابته زي ملفات ال BIOS ال بيكون فيها معلومات عن ال Motherboard وال Hard disk وال Ram ونوع جهازك ومين ال يعمل Boot للجهاز وسنه تصنيع الجهاز كل دي معلومات لازم جهاز أول متيجي تشغله لازم تعدي على ال BIOS Files ...

- فلذلك مصنف خطورته كبيره النوع دا لأنك لسه بتشغل الجهاز فالاول وانت بتشغله راح يعمل Check على ملفات ال BIOS لقاه Infected ب Rootkit فكل الجي ال Rootkit هيعمله Infected وكمان ال Tools صعب عليها تكتشفه .

- رابع نوع عندنا من أنواع ال Malware وهو ال Book kits ودي بتختلف عن اللى فاتت وهي ال Root kits فطريقه ال Installation

بتعتها وطريقه عمل ال Control على ال Operating System ...
ال Book kits دي بتعمل Attack عال Operating system
بتاعك من قبل ميشتغل أصلا فشوف الخطورة وصلت لأيه انك من قبل
متعمل اي حاجه ! .

- خامس نوع معانا وهو ال Trojans وبرضه شكل من أشكال ال
Malware واللى بيميزه أنه واخد شكل ال Legitimate
software بمعنى شكله زي ال Software الحقيقي اللى عندك تماما
فأنت مبتاخدش بالك منه وبيبقا موجود فالمواقع على شكل ملفات تحميل
لكراكات وغيره ... فأنت تنزله فيبتدي ياخد Un Authorized
Access لجهازك وانت عادي مطمن للملف !! وتلاقيه بينزل
Software أخر على جهازك وهكذا ... الفكره فال Trojan انه
بيخدعك أنك تنزله وتطمن ليه وبتلاقيه واخد نفس ايقونه البرنامج اللى
عاوز تنزله فتبدء تنزله وتشغله ب Double click وبعد كدا بياخد
صلاحيات للوصول ل Sensitive data على جهازك وهكذا ... فمثلا
انت نزلت ملف ل game معينه وليكن من موقع مشبوه أو عادي
ومفيش اي اختلاف وضحلك ان الملف دا بيحتوي على Trojan ...
والسطور البرمجيه المكونه لل Game دي مبينهم كام سطر
malicious ل trojan فأنت اما تيجي تشغل ال game هتشغل
معاها ال Trojan الموجود فنفس الملف لما تعمله ال execute !! .

- سادس نوع معانا وهو ال Back Door هو شكل من أشكال ال
Software أو ال Software معموله تعديل أو Modification
بحيث يساعد ال Attacker فجزء ال maintaining Access بعد
اما يعمل Exploit لل Target فال Attacker عاوز ينشأ Back
door عند ال Victim machine عشان لما يعوز يرجع لل
Victim تاني يرجع منه ميقعدش يعمل خطوات ال Penetration
testing من بدايتها أو مثلا ال Vulnerability اللى دخل منها ال
Attacker اتعملها patching يعني اتقفلت أو اتعملها ترقيع فكدا

الدنيا باظت بالنسبه لل Attacker فيروح يزرع Back door عند ال Victim عشان لما يحتاجه قدام يلاقيه ... وكمان لو ال Attacker زرع ال Back door عند ال Victim مبيحتاجش انه كل ميجي يدخل لازمله ال Username وال Password ...

- لاء عن طريق ال Back door دا بيقوم داخل عادي ... ولو انت متابع هتلاقيني شرحت الكلام دا بالتفصيل الممل فملف ال Network Security فال Post Exploitation phase ووضحت كل نقطه فيه هتلاقيه فالبروفايل عندي أرجعله عشان تفهم منه mentality ال Attacker شغاله ازاي وتحط نفسك مكانه عشان تعرف عاوز يوصل لأيه وتوقفه .

- معلومه كدا ... ال RAT اللى هو ال remote Access trojan مشابه جدا لل Back door فعمله ونفس ال features بتعته ... ودا النوع السابع عندنا بالمناسبه من أنواع ال Malware وهو ال RAT ودا عباره عن remote malicious Administration tool بتسمح لل Attacker انع ينفذ Commands بطريقه remotely عند جهاز ال Victim اللى أتعمله أختراق ... ودايما ال RAT بيكون على شكل Client- Server Model بمعنى ال Attacker بيكون عنده نسخه ال Server من ال RAT وعند الطرف الأخر ال Victim بتلاقي ال Client وبيبدء ال Attacker يعمل Connect من ال Server بتاعه لل Client الموجود عند ال Victim machine ويبعتله Commands ينفذها عنده زي مقولنا Remotely ... فممكن يعمل create ل User أو يمسح ملفات أو يعمل Upload لملفات malicious عند ال Victim Machine وهكذا .

- عندنا الشكل التامن وهو ال Spyware ودا عباره عن Malware عند جهاز ال Victim يعمل تجسس على ال User Activity وليكن أو يعمل Collection ل Information عن ال User ويشوف ايه

المواقع اللى ال User دا زارها وطبعا كل دا بدون مال Victim يعرف حاجه ... ومش هتلاقي ال Spyware جايلك منفرد لوحده كدا هتلاقيه جايلك معاه Rootkit أو Trojan مدموج معاه ال Spyware وهكذا.

- النوع التاسع عندنا وهو ال Botnets ودا أختصارا لكلمتين وهما ال Robot Network ودا عباره عن Collection من الأجهزة اللى تم أختراقها بالفعل وأصبحت تحت تحكم ال Attacker حاليا وكلها ضمن شبكه واحده وهي ال Botnet بيستخدمهم ال Attacker مثلا ف Attack زي ال DDOS Attack ... فتلاقيه أخترق جهاز A وB وC فيحطهم كلهم فشبكه واحده تحت تحكمه ويوجهلهم ال Commands اللى عاوز ينفذها عن طريق ال C&C Server اللى هو ال Command and control Server عشان يوجه الاجهزة للى هو عاوزه .

- النوع العاشر معانا وهو ال Ransomware ودا جي من كلمتين وهما ال Ransom ومعناها فديه وال Software برامج ... يبقى برامج الفديه ... ودا منتشر بشكل كبير فالفتره الاخيره لانه بيطلب فديه أو مقابل مادي من ال Victim عشان يفكله ملفاته اللى شفرها وبتدفع لل Attacker بالعمله المشفره Bitcoin طبعا عشان الطرفين محدش يتتبعهم وبيديك فتره معينه علشان تدفع المبلغ المطلوب منك مقابل انه يرجعلك ملفاتك ويديك مفتاح فك تشفير ملفاتك ولو مدفعتش بيمسحلك ملفاتك ودا نوع خطير خصوصا لو وقع على Sensitive data لمؤسسه ما أو بنوك أو ما شابه .

- النوع الأخير معانا من أشكال ال Malware وهو ال Information Stealers ودا بيسرق ال data ال Private عندك زي ال encryption keys أو login credentials أو Credit card data وليه كذا شكل منه ال Keylogger ...

- ودا بيسرق الكلام بتاع ال user اللى بيكتبه عال keyboard عن طريق انه بيسجل ضغطات ال user عالكيبورد فلو فتح موقع هتلاقيه سجله ولو كتب username و password هتلاقيه برضه بيسجله وهكذا ... فيه شكل تاني وهو ال Screen Recorder ودا بياخد سكرين أو لقطه من ال Active window اللى فاتحها ال Victim عنده حاليا وكمان بيعمل Screen recorder للشاشه الخاصه بجهاز ال Victim ... بالاضافه الى انواع من ال information stealers بتفتح ال Webcam وتاخد لقطه لل Victim او المايك الداخلى الخاص بجهازك وتسجل لل Victim أو تتنصت عليه وتسمع مكالمته وهكذا ... عندنا نوع أخر وهو ال Ram Scraper ودا من اسمه بيزحف عالرامات ويسرق ال Data الموجوده فيها اللى بتكون Decrypted عشان اي Data فالرام بتكون جاهزة عالتنفيذ بتلاقيها بشكلها الخام ...

- زي مثلا ال Victim فتح موقع بنك معين من جهازه وفيه معلومات حساسه دخلها فال Browser المعلومات دي بتتخزن فالرام بشكل decrypted غير مشفره فيقدر ال Attacker يروح يسرقها وبتكون ال data دي encoded بيقوم ال Attacker واخدها عملها decoded ويطلع منها ال Sensitive data اللى هتفيده ودا بيحصل عن طريق ال malware اللى من النوع information stealer ...

- وطبعا عندنا أنواع أخري من ال malware لم يتم ذكرها زي ال Adware وال greyware وال scareware وال fake ware لكن هنا مش محور حديثهم ولا هما مختلفين كتير عن اللى ذكرناهم وبكدا نكون اتعرفنا على انواع ال malware المختلفه عشان نفهم فالجزء الجي باءذن الله ازاي ال malware بيوصل للأجهزة ويصيبها .

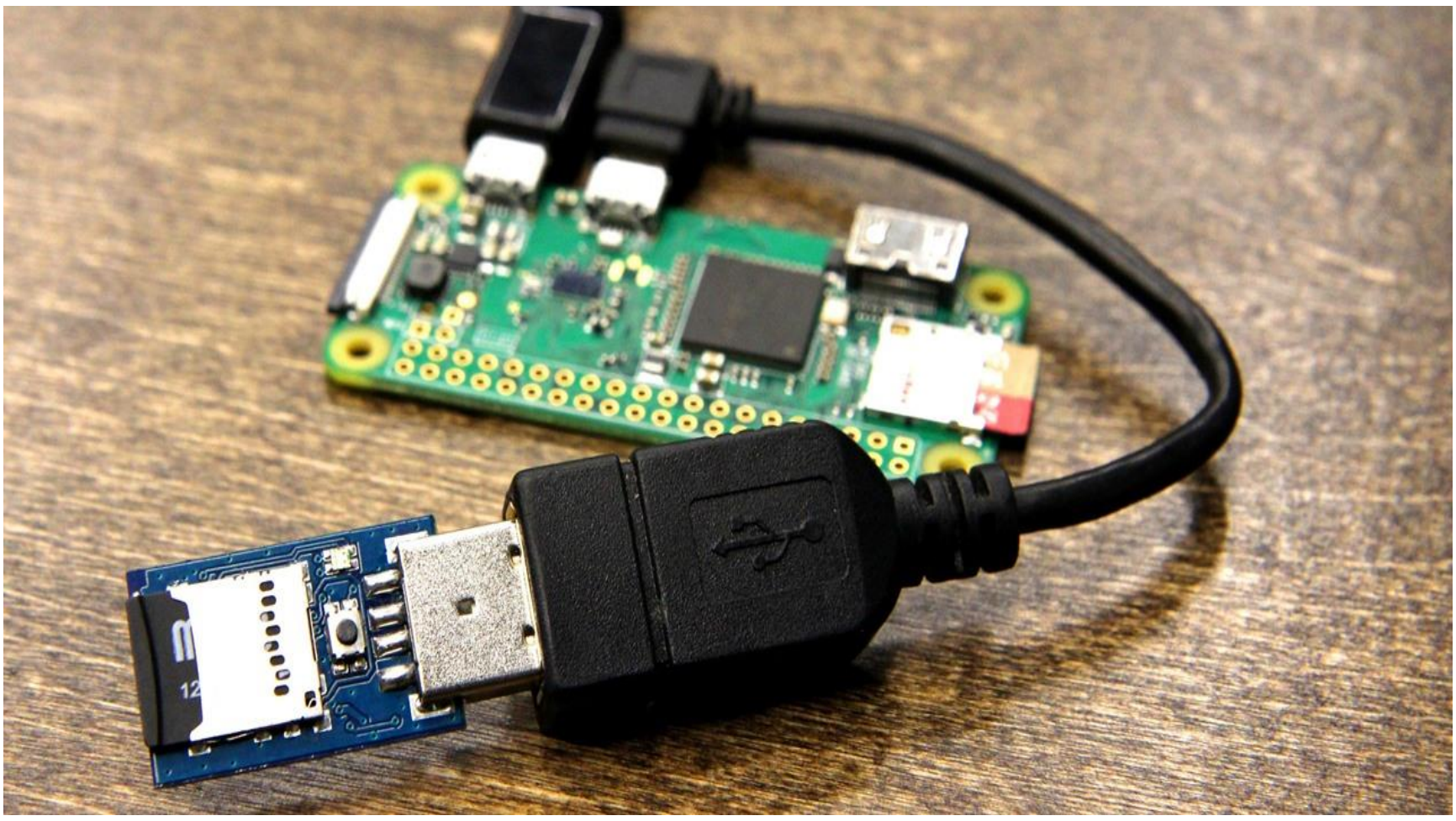
3.Malware Delivery:

- هنتكلم فالجزء دا عن ازاي ممكن ال Malware ممكن يوصل لل End point ويصيب جهاز ال Victim ... عندنا كذا technique ممكن ال Malware يستغلهم عشان يوصل لل Victim machine زي ال Physical media وال email attachment وال URL وال links وال Drive by Downloads وال Web Advertising وال social media وال File Shares وال Software وكل طريقه منهم هناقشها ونفصصها Vulnerabilities ... .

- أول حاجه معانا وهي ال Physical Media ... خدبالك من نقطه وهي ان ال Malware عشان تستخدم طريقه ال Physical media عشان تنتشر مبيبقاش عندها اي طريقه تانيه ... يعني ولا طريقه نفعت عشان كدا لجأت للطريقه دي عشان تنتشر عند ال Victim ... زي ال Malware اللى بيبقا على ال USB كدا تماما هتلاقيه فشل انه يبعتهولك بأي طريقه تانيه عشان كدا لجأ لل USB ... فتلاقي ال Developer اللى صمم ال malware وحطه عال USB مبرمجها انك أول متحط ال USB فالجهاز تشتغل علطول يعني عملها Autorun عشان تروح تعمل infect بال malware لل Boot Sector عندك فالنظما اللى نظام الاقلاع اللى بيقوم النظام عندك ... وعندك حاله تانيه برضه ال Developer اللى صمم ال malware مش شرط يخليه يشتغل بطريقه Autorun لاء عادي ممكن يسيبه بحيث ال user اللى يشغله من ال USB وال malware عامل infect لملف معين فال USB فمجرد مال User يشغله بس هيحصل نفس القصه برضه .

عندنا أجهزة physical بتيجي عليها ال malware أو انت ممكن تضيف اللى انت عاوزه اللى هي HIDS اختصارا ل Human Interface Devices ... .

- زي ال Rubber Ducky وال USB drive by اللى هي كابلات شحن الموبايل فيه انواع منها مخصصه لسحب ال Data أو انها تصيب الاجهزة اللى هتتصل بيها بال malware لما توصلها عشان تشحن الجهاز ... تعالى نشوفهم .



- ال Tools دي لما بتوصلها بجهاز ال Victim هي متبرمج جواها Scripts جاهزة بتعمل execute لبعض ال Commands فبتخلى ال Malware يشتغل عندك عال Victim System ... وفيه منهم أنواع مصممه انها تتخطي ال Antivirus وتعمله Bypass وتعمل infect لل System .

- تاني شكل معانا وهو ال Email ودا أشهرهم واللى بيجي من خلاله ال Malware ويصيب الأجهزة عندك فالمؤسسه ودا بسبب ال Poor Security Awareness قله وعي الانسان بالاحتياطات اللى ياخدها عشان ميقعش فالفخ وينزل Attachment من email ويكون Malicious ويصيب ال Endpoint وفيما بعد وينتقل للأجهزة الاخري ... وال Attacker دايما بيلعب على حته الفضول عند الموظف أو ال User انه يحطلك رساله تشد فضولك انك تفتحها وهو مش عاوز منك الا كدا !! زي مثلا الحق السحب على سياره وكن من أول الرابحين وهكذا من الرسايل اللى مبنيه على فضول الانسان انه يتجه لاكتشافها فخد بالك من ال emails اللى بتجيلك كويس وحاول تشتغل وتركز وتصرف على ال security Awareness للموظفين بتوعك زي مبتصرف عليهم فال Tech والكورسات !! ودا كله بيندرج تحت هجمات ال Social Engineering اللى منها ال Phishing اللى بتم من خلال ال Email ... ال Attacker ليه بيستخدم ال Phishing أو يبعت لل Victim اللى بيترجته عن طريق ال Email !؟ حط نفسك مكانه انت ليه تقعد ايام وشهور عشان تحاول تدخل لل Network من خلال ثغره والله أعلم هتلاقيها ولالاء وشوف كم المجهود اللى هتاخده منك عشان تعمل للترجت exploit و post exploitation مرورا بأني عاوز اعمل Bypass لل Firewall فليه كل المجهود دا !! طب منا هروح أعمل Crafted email وشكله حلو ميتشكش فيه من تجاه شخص معندهوش Security Awareness خالص !! وهدمج فال email دا malicious file أو Link ل Malicious file واستنى ال Victim لحد ميفتح ال email وينزل ال Attachment وبكدا أكون وصلت للى أنا عاوزه بسهوله عن طريق ال Social engineering بحاجه زي ال email phishing ووفرت على نفسي الوقت والمجهود بتوع ال exploitation وال Bypass اللى ذكرناهم وكمان متضمنش هينجحوا ولالاء ... ودي طريقه بتنفع فالشركات اللى صارفه كتير على ال Technology بتعتها ولكن الموظفين عالله خالص فالحته دي .

- الشكل التالت معانا وهو ال URL Link ودا عباره عن لينك ممكن يجيلك فال email ومش شرط برضه ممكن يجيلك عن طريق اي وسيله تواصل تانيه زي Messenger مثلا ويقولك لينك لمباره كرة القدم كذا ضد كذا وهيبقا عامل Social engineering وعارف كويس ان التوقيت دا مميز وال Attacker هيلعب على فضولك فيه ... وانت ك Victim لو ضغطت على ال Link دا هيعملك Redirect ل Website بالفعل موجود عليه ال Malware وبيتم تحميل ال Malware دا مجرد متزور الموقع فقط لاغير ... ممكن ال URL link دا يجيلك فال email زي مقولنا وممكن يجيلك فال Documents أو ال Attachment زي ملفات ال Word أو ال Excel وهكذا وممكن يجيلك برضه زي مقولنا عن طريق ال App Message زي الواتساب وغيره وكمان من مواقع ال Social media الأخري... وكمان عندك ال URL Short link اللى هو فعلا هتلاقي ال Link بتاع ال Malware بس معموله Short من خلال المواقع المتخصصه فذلك زي Short linker مثلا عشان ال Victim ميشكش فيه وعندك مواقع زي Virus total ممكن تفحصلك ال Links دي قبل متدخل عليها .

- الطريقه الرابعه عندنا هي ال Drive By Downloads ودي معناها ان ال Malware بيتعمله Install تلقاءي عندك عالجهاز الخاص بيك أول منتا ك User بتزور ال Site بتاعك ... ودا أخطر لأن ال Victim مش محتاج يعمل Click على أي Link فال Web page ودا بيستغل أي ثغره موجوده عندك فال Browser زي مثلا لو ال User بيستخدم Windows 7 شغال عليها نسخه من برنامج ال Flash اللى كان فيها ثغرات ومتعملهاش Patching هتلاقي ال Malware نزل عندك بدون متضغط على Link مجرد فقط زيارتك لل Web page المصابه هتلاقيه بدء ينزل عندك تلقاءي ...

ـ فدا زي موضحنا بيبقا عيب فال Browser عندك بيبقا فيه ثغره ال Malware دا بيستغلها فأنك أول متزور الصفحه دي يبدء ينزل عندك تلقاءي ... وفيه طريقه تانيه ان ال Attacker دا بيشتري Specific Domain مشابه ل Main domain وليكن عندك Google.com دا محجوز ... هتلاقيه يروح يشتري Google Network.com حاجه كدا مشابه ل Google عشان يستغلها فال Social engineering ... ويرفع عال Domain ال Malware بتاعه ويستنى ال Users انهم يدخلوا عليه أو يبعته لحد مثلا يستهدفه بيه وكمان هتلاقيه بيضبط ال SEO اللى هي ال Search Engine Optimization اللى هي الكلمات البحثيه اللى ال Users يكتبوها فيوصلوا للموقع بتاعه وكمان ال SEO مسؤله عن انها تظهر ال Domain بتاعك أو الموقع فأول نتايج البحث فحاله ان حد بحث عنها وهكذا بحيث يزود فرص ظهور الموقع بتاعه لأكبرعدد فيدخلوا للموقع اللى عامله ال Attacker على اساس انه حقيقي وينزل عندهم ال Malware بشكل تلقاءي زى مذكرنا ... وفيه عندنا طريقه تانيه وهي ان ال Attacker يبدء يترجت موقع معروف الناس بتزوره دايما زي مثلا موقع الجامعه وليكن فيه ألالاف من الطلاب بيدخلوه يوميا فال attacker يقوم مخترق الموقع دا بطريقه ما عن طريق انه يبحث فيه عن ثغرات الويب اللى مصاب بيها ويبدء يعملها Inject وبعد ذلك يرفع ال Malware بتاعه على الموقع دا وبكدا عرض كل ال Users اللى بيدخلوا للموقع انهم يحملوا ال Malware دا فشوف انت عدد ال Victims اللى هينزل عنده ال Malware دا ... والطريقه دي بنسميها ال Watering Hole Attack .

ـ **الطريقه الخامسه** معانا وهي ال Web Advertising وليه اسم تاني وهو ال malvertising ودا ال malware بيوصلك من خلال ال Online ads اللى هي الاعلانات يعني ...

- ال Attacker بيروح يعمل اعلانات مدفوعه على صفحات ومواقع معروفه بحيث تظهر لأكبرعدد من ال Victims ومجرد متضغط على الاعلان انت هتلاقيه عملك redirect لل Webpage اللى عليها ال Malware وهتلاقي ال Malware بدء ينزل عندك تلقاءي وممكن من غير متدوس على اي اعلانات كمان الجهاز بتاعك يتصاب بال Malware ودا عن طريق الExtension اللبتنزلها فالمتصفح عندك ... فال Attacker يستهدف Extension فيه عدد كبير بيستخدمها ويقوم زارع ال Malware بتاعه فيها وكل ال Users اللى نزلوا ال Extension دي هيصابوا بال Malware .

- الطريقه السادسه معانا وهي ال social media ودي منتشره بشكل كبير فالفتره الحاليه عن طريق ال الاعلانات وشويه Social Engineering معاهم ويقنع ال User انه يضغط على الاعلان أو ال Link اللى برضه بيكون حاطط فيه malware ومستنى ال User يضغط عليه بس الفكره في أقناعه بكدا فقط ...

- الطريقه السابعه والأخيره معانا وهي ال Software Vulnerabilities ودي عباره عن وجود ثغره أو نقطه ضعف عندك فال System أو ف APP شغال عندك علشان انت معملتش update للنظام عندك أو المتصفح أو الابليكشن ... عندك مثال وليكن ال Buffer Overflow زي ال Stack Overflow أو ال Heap Overflow والاتنين موجودين فال Memory بتاعت جهازك ... وال Stack دا المساحه الموجوده فال Memory بتعتك علشان زي مقولنا أي حاجه هتتنفذ على جهازك لازم تروح تاخد مكان فال memory بعد اما يتعملها execute فبتاخد مكان فال Memory وبعدين تتنفذ من خلاله وال Stack دا ليه مساحه معينه ... فبيقوم ال Attacker مالى ال Stack دا زياده عن حجمه فيسبب ال Overflow وال System عندك يحصله Crash ودا راجع انك بتستخدم function مصابه ...

بال Buffer Overflow زي ال Strcpy دي مثلا داله لو لقتها مكتوبه عندك فالكود أعرف انه معرض لل Buffer Overflow Attack لانها مبتحددش Limit للقيم اللى بتدخلها فال Stack فأنت تقدر تدخل أي قيمه وتسبب Crash لل System ... أثناء مال Attacker عمال يضيف قيم زياده لل Stack بتعتك هتلاقيها عماله تتضغط وهيحصل Crash زي مقولنا وسط كل دا ال Attacker هتلاقيه بيزرع ال Malicious code بتاعه وسط ال values اللى بيضفها لل Stack عندك جوا ال Memory .

ـ أما بالنسبه لل Heap Overflow فهو انه ال Heap دا برضه عباره عن مكان جوا ال memory بيتخزن فيه ال Addresses اللى بتوصلك لل APP وليكن أو الخاصه بتنفيذ Processes معينه ... هتلاقي ال Attacker بيروح جوا ال Heap ويغير العنوان الموجود الخاص بال APP معين بالعنوان بتاع ال malware بتاعه فأنت تيجي تروح ل App تلاقي نفسك بتتوجه للمكان اللى فيه ال Malware تلقاءي ... وبس كدا دول أشهر طرق ال Malware Delivery اللى ممكن ال malware يتنقل بيهم من خلال ال Attacker لل Victim ... وطبعا فيه طرق تانيه ولكن دول أشهرهم زي مقولنا وفالجزء الجاي باءذن الله هنعرف مع بعض ازاي ال Malware بيعمل Evasion لل defensive tools اللى عند ال User زي ال Antivirus .

---

4. Malware Evasion Techniques:

ـ هنشوف مع بعض ال Most Common techniques اللى بيستخدمها ال Malware عشان يعمل Bypass لل defensive Tools اللى عندك .

ـ الفكره من ال Malware مش انه يعملك Bypass لل Defensive Tools اللى عندك فال Network لاء ... هدفه انه عمليه ال Exploitation تنجح بالفعل عند ال target ويكمل فال post Exploitation Techniques عشان اهداف ال Attacker أكبر من كدا و اللى منها وهيستخدمها ال Attacker زي ال Steal Credentials وال Data Exfiltration وغيره .

ـ خد بالك الطرق اللى هنذكرها هنا لل Evasion Techniques مش كلها لأن ال Researchers كل يوم بيكتشفوا طرق جديده من ال Attacks اللى بتحصل فلازم انت كHunter تكون Updated دايما بالطرق دي وبال Malware الجديده وطرق ال Bypass ليها علشان تبقا Updated أول بأول .

ـ أول طريقه معانا فطرق ال Evasion لازم تكون عارفها ك Huter هي ال Alternate Data Streams ... بنسميها ال ADS كأختصار وهي عباره عن Feature في نظام ال Windows في نظام الملفات NTFS ومش موجوده فأنظمه ال FAT أو اي نظام تاني ... تعالة نفهم ايه ال NTFS وال FAT .

ـ دول عباره عن أنظمه أو طرق لتخزين الملفات والتعامل معاها فال System اللى هو ال Windows ... ال NTFS دي اختصار ل New technology file system دا بأختصار الأحدث والاسرع في تخزين الملفات والتعامل معاها و More Secure عن ال FAT اللى هو اختصار ل File Allocation Table ودا الاقدم وبنستخدموه مع كذا نظام تشغيل زي ال Windows وال Linux بس مش Secure زي ال NTFS وكمان مبيدعمش ال Feature بتعتنا اللى هي ال ADS بالاضافه الى ان ال NTFS دا مخصص لنظام Windows فقط على عكس ال FAT ...

- وال FAT تستخدمه فحاله انك ك User بتستخدم فلاشه أو كارت ميموري لأن كل الأجهزة بتقرء ال FAT انما ال NTFS لو بتخزن ملفات أكبر من 4 جيجا وعاوز سرعه وأمان أكتر لل Data وكمان بتستخدم نظام Windows فهيكون دا الانسب ليك فالحاله دي ... بكدا نكون عرفنا الفرق مبينهم وحالات استخدام كل واحد منهم ... نرجع بقا للموضوع بتعنا فأول Evasion technique لل Malware وهو ال ADS .

- طب ازاي بتشتغل ال ADS ؟؟ عندنا كل ملف فال NTFS عنده ستريم اساسي ال main data stream ودا المكان اللى بيتخزن فيه المحتوى الطبيعي للملف ... ال NTFS بيسمحلك انك تضيف للملف Streams أضافيه بتبقا Hidden بدون ميأثر على حجم الملف اللى ظاهر لينا ... ناخد مثال عشان المعلومه توصل .

- لو عندك ملف اسمه File.txt فممكن نخزن جواه Stream أضافي مخفي من غير ميبان فالحجم الفعلى للملف بتعنا ... عن طريق الامر دا على سبيل المثال اللى قدامك فالصوره ... توضيح بسيط ال Stream يعني نقدر نخزن Data أو Files مخفيه جوا ملف عادي عال System من غير ميحصل اي تغيير فحجم الملف الاصلي عندك عالنظام ... ودا اللى ال Malware بيستغله عندك ان مثلا ال User يروح يفتح ال File اللى اسمه File.txt اللى جواه welcome مثلا على اساس ان جواه الكلام دا فقط بس هو جواه كلام تاني معموله Hidden من ال Attacker عن طريق انه استغل ال Feature اللى هي ADS فال NTFS فال Windows وضاف Data تانيه hidden جوا ال Stream الخاص بالملف دا ... تعالى نرجع للمثال .

```
echo “This is not ADS” > file.txt
echo “This is in ADS” > file.txt:stream1
```

- هتروح عندك فال CMD فال Windows وتحط ال Command الاول دا اللى هيعمل Create ل File بالاسم file.txt وهيحط فيه الكلام دا عادي اللى هو This is not ADS ودا كدا بالنسبالنا File عادي وبعد كدا نروح ننفذ ال Command اللى بعده اللى هو خاص بالجزء اللى بنتكلم فيه ال ADS وال Stream لل File وتضيف الكلام اللى قدامك دا This is in ADS وبعد ذلك هتروح تعمل ال Command اللى هو dir عشان تشوف ال Files اللى عندك هتلاقيهم بنفس الحجم زي مكنا قولنا ولو عملت ال Command ال Type للملف نفسه هتلاقي المطبوع قدامك ال Content اللى هو This is not ADS طب أومال راح فين ال Data اللى ضفناها فتاني سطر !!؟ هو دا اللى بنتكلم عليه من خلال ال ADS تقدر تخفي data جوا File معين من خلال خاصيه ال NTFS الخاصه بملفات ال Windows ... نشوف الكلام دا بشكل عملي .

```
C:\Users\s>echo "this is not ADS" > file.txt

C:\Users\s>echo "this is ADS" > file.txt:stream1

C:\Users\s>
```

```
 Directory of C:\Users\s

02/11/2025  11:54 AM    <DIR>          .
02/11/2025  11:54 AM    <DIR>          ..
06/17/2022  03:00 PM    <DIR>          .android
08/20/2024  04:43 AM    <DIR>          .cache
03/13/2022  03:09 AM    <DIR>          .idlerc
01/07/2024  07:30 AM    <DIR>          .Ld2VirtualBox
01/07/2024  11:57 PM    <DIR>          .swt
11/09/2024  03:28 AM    <DIR>          .vscode
10/27/2024  06:23 PM    <DIR>          3D Objects
07/07/2022  03:34 PM                26 ahmed.py
10/27/2024  06:23 PM    <DIR>          Contacts
02/11/2025  10:24 AM    <DIR>          Desktop
12/31/2024  02:09 PM    <DIR>          Documents
02/11/2025  11:53 AM    <DIR>          Downloads
03/06/2022  12:23 PM    <DIR>          Favorites
02/11/2025  11:54 AM                20 file.txt
03/06/2022  12:23 PM    <DIR>          Links
07/24/2024  01:14 AM    <DIR>          Music
01/18/2022  07:55 PM    <DIR>          OneDrive
12/31/2024  12:20 AM    <DIR>          OneDrive - BUC
12/27/2024  09:28 AM    <DIR>          Pictures
07/29/2022  04:35 PM    <DIR>          PycharmProjects
03/06/2022  12:23 PM    <DIR>          Saved Games
07/01/2022  05:10 PM                53 Search.code-search
03/06/2022  12:23 PM    <DIR>          Searches
12/17/2024  08:45 PM    <DIR>          source
11/15/2022  02:27 AM               241 Untitled-1.py
01/17/2025  08:00 PM    <DIR>          Videos
               4 File(s)            340 bytes
              24 Dir(s)  72,508,297,216 bytes free
```

ـ لو انت Programmer أو بتعرف تكتب كود فالعموم فالجزء دا هيفديك ... عندنا Code بال C بيشرح الفكره اللى قولناها دي تعالى نشوفه ونحلله مع بعض لو مهتم تعرف تفاصيل .

```
#include <windows.h>
#include <stdio.h>

void main() {
        ...
        hStream = CreateFile("file.txt:stream2",
                GENERIC_WRITE,
                FILE_SHARE_WRITE,
                NULL,
                OPEN_ALWAYS,
                0,
                NULL);
        if(hStream == INVALID_HANDLE_VALUE)
                printf("Cannot open file.txt:stream2\n");
        else
                WriteFile(hStream, "This data is hidden in the
stream. Can you read it???", 53, &dwRet, NULL);
}
```

ـ الكود دا مش بيكتب فالملف العادي بتعنا اللى هو file.txt لاء... بيكتب فجزء مخفي وهو ال Stream اللى مسمينه Stream2 ... وزي مقولنا ال Size الخاص بالملف هيظهر عادي جدا لكن جواه data مخفيه مش هتشوفها الا بطرق معينه ... زي انك من خلال ال CMD تكتب ال Command ال More عشان تعرض محتوي ال Stream المخفي ... ال Library اللى هي Windows. H دي بتحتوي على ال Functions اللى بتتعامل مع الملفات في نظام Windows ... وعندك ال Create file دي عباره عن Function فال Windows علشان تعمل Create لملف جديد فال Windows .

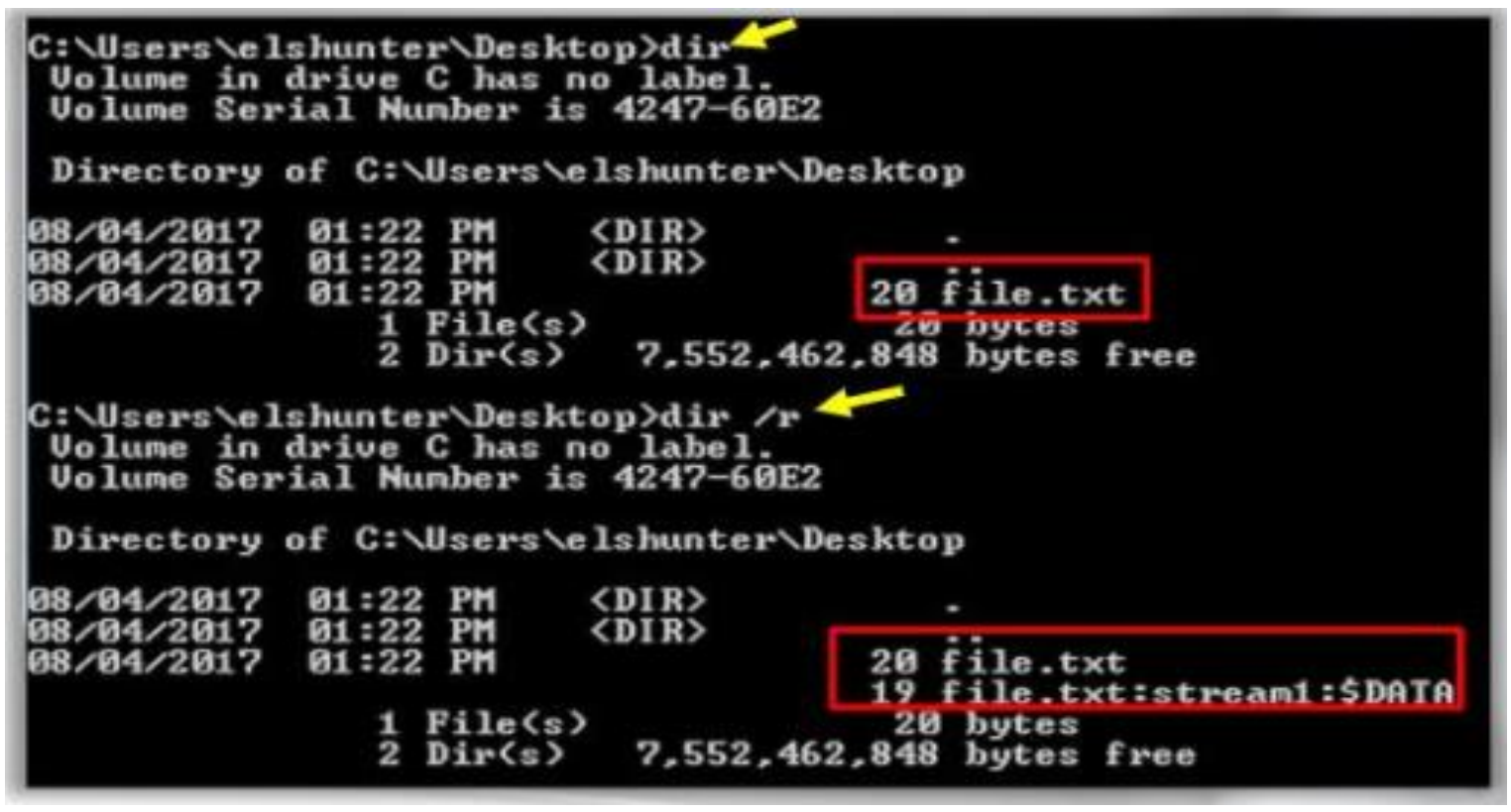
ـ وبعد ذلك هنفتح ال Stream 2 اللى هو الملف المخفي اللى جوا ال File الاصلي بتعنا زي مكنا شرحنا فوق ... بعد كدا بتعمل ال Setting للملف زي Generic-write اننا نسمح بكتابه ال Data فملف واحد ... ال File-share Write دا بيسمج لبرامج تانيه بالوصول للملف بتعنا لما نشغله ... ال Open always معناها لو الملف موجود أفتحه ولو مش موجود اعمل Create ملف وافتحه .

ـ وطبعا عندك ال hStream دا ال Variable اللى زي المفتاح المسؤول عن فتح ال Stream جوا الملف ونقدر من خلاله نقرء ونكتب ونعدل فالكود براحتنا .

ـ بعد كدا عندك ال If Condition اللى هي بتقولك Invalid handle value بمعنى لو ال Function بتعتنا اللى هي create file() فشلت في فتح أو انشاء ملف ... فلو فشلت تطبعلنا عن طريق ال Printf السطر دا ("Cannot open file.txt:stream2\n") .

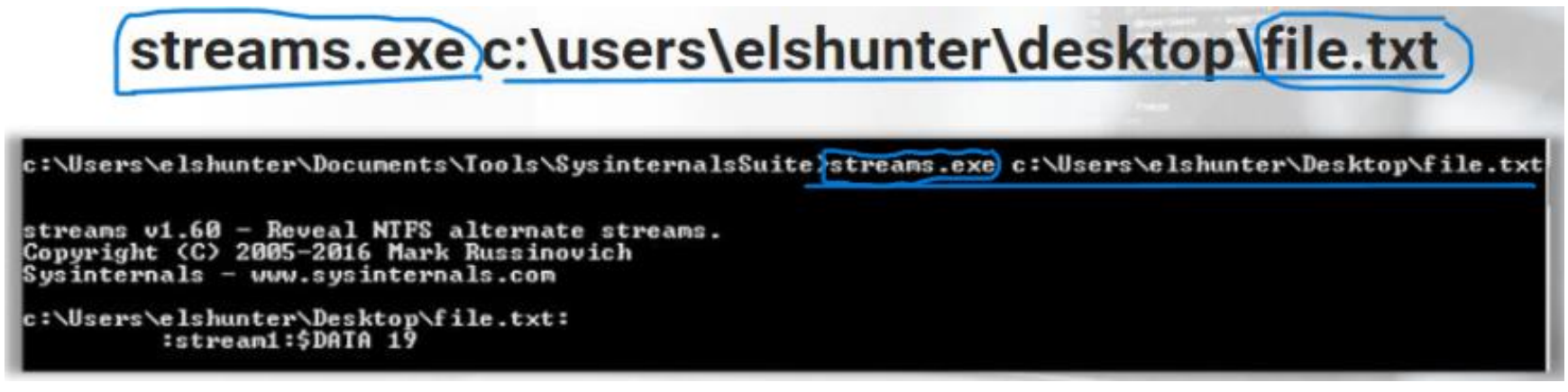
ـ الخطوه اللى بعد كدا عاوزين نكتب ال Data بتعتنا جوا ال Stream المخفي ... عن طريق ال Function بتعتنا اللى هي Write file() عشان نكتب ال data بتعتنا جوا ال Stream بتعنا ... ودا السطر اللى هنكتبه "This data is hidden in the stream. Can you read it???" والكلام دا هيتم تخزينه فال Stream2 زي موضحنا ... وعدد الأحرف اللى هنكتبها 53 وعندك ال&dwRet دا عباره عن Variable هيحفظ عدد الأحرف اللى هيتم كتابتها بنجاح وال Null يقصد بيها مش هنستخدم أي اعدادات اضافيه ... طب ايه اللى هيحصل لو شغلت الكود!!

ـ هيتم انشاء ملف اسمه file.txt والملف هيظهر عادي ومفيهوش اي حاجه غريبه ولكن جواه فالخلفيه عندك ال Stream مخفي على اسم ال Stream2 وال data اللى هيتم كتابتها فيه متخزنه ولو فتحت الملف بأستخدام ال Notepad مش هتشوف ال Data دي فلازم تستخدم ال Commands الخاصه زي More ... أو ال Command ال dir ومعاه ال /r دا هيطلعلك ال hidden Stream file .

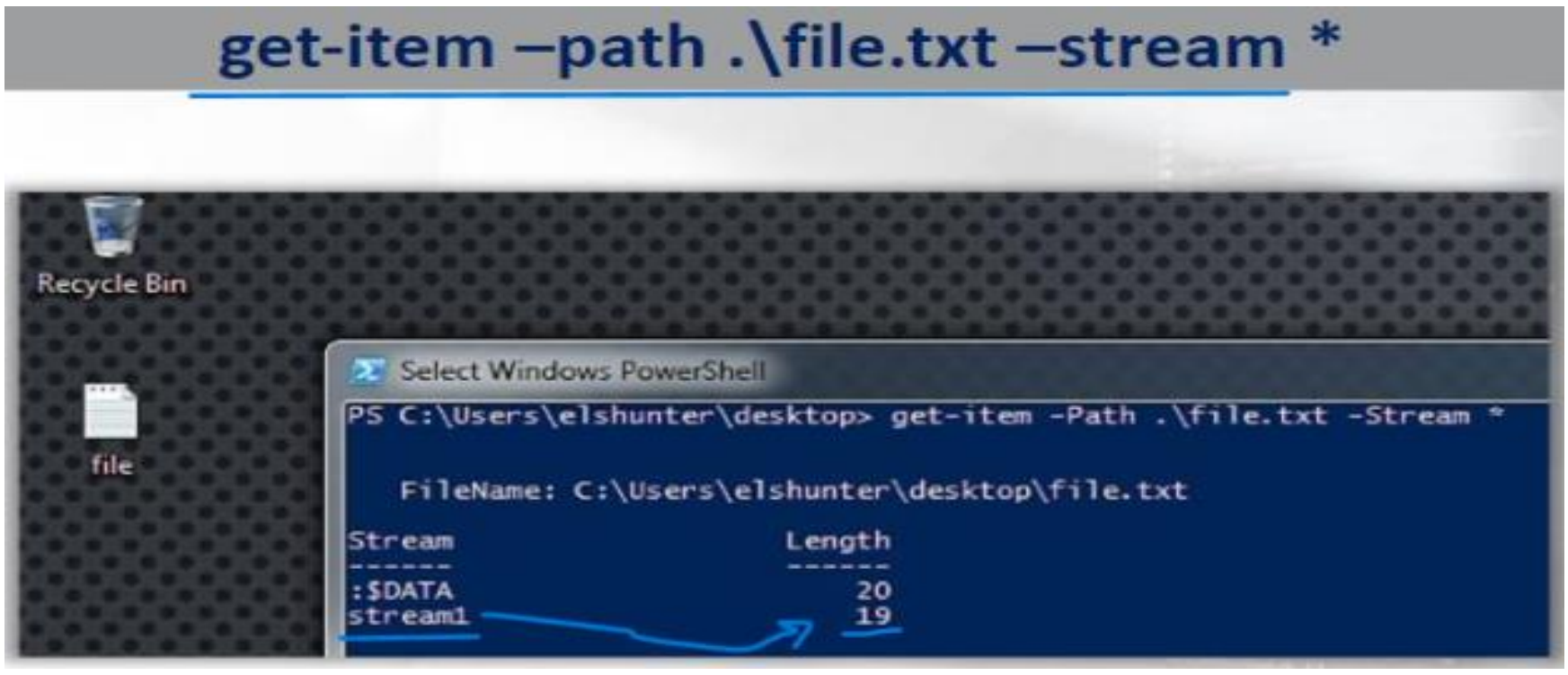


- وبرضه عندك ال Command ال type وال More لو استخدمتهم كدا هيطلعولك المحتوي الطبيعي لل File ولو استخدمت معاهم ال stream: هيجيبولك المحتوى المخفي ولكن مع ال type هيدوك Error لكن مع ال More هيدوك النتيجه عادي .

- وعندك بعض ال Automated tools زي ال Streamed كدا تقدر تنزلها ودي من ضمن الحزمه بتاعت Microsoft اللى اسمها ال Sys internals ... دي بتقدر تجبلك ال Stream data اللى مستخبيه جوا ال Files عندك عال System ومعمولها Hidden ... تدي ال Tool اسم ال File وال Path الخاص بيه وهي هتجبلك ال hidden data .



ـ برضه عندك Tool تانيه وهي ال get-item ودي من ضمن ال مجموعه من الادوات اسمها ال Command let اللى هي اختصارها Cmdlet ودي عباره عن PowerShell tool تقدر تقوملك بنفس الوظيفه وتطلعلك ال Stream data بشكل أسرع وهتطلعلك ال ADS information بشكل أكفء من ال Tools أو ال Commands اللى ذكرناها ... هتحتاج بس تديها ال Parameter اسمه ال – stream وتديها ال path الخاص بالملف بتاعك وهي هتشتغل .

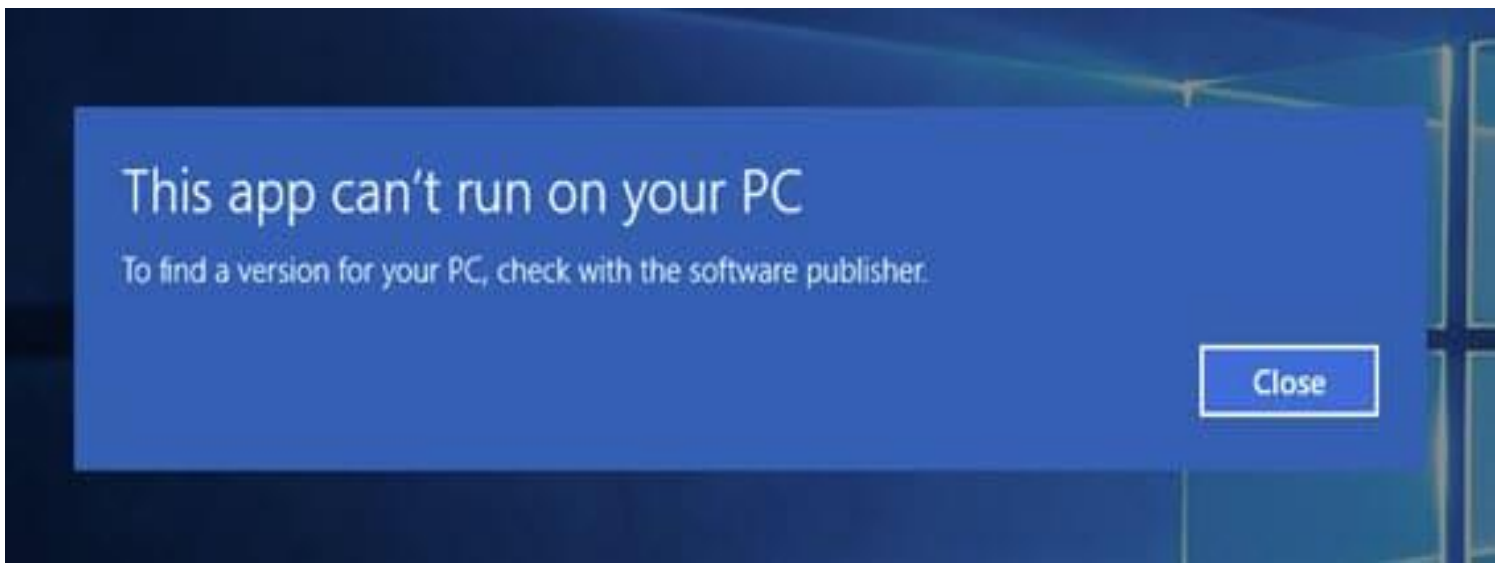


ـ ال ADS شركه Microsoft بتستخدمها بشكل اخلاقي مش هيسبب ضرر للمؤسسه عشان تخزن فيها ال Zone Identifier طب ايه دا !؟... تعالى ناخد مثال يوضح الدنيا ... مش حضرتك دلوقتي عاوز تنزل ملف ما فعندك كذا طريقه زي انك تنزله من جهازك أو من Network انت موجود فيها يعني Local أو من ال Internet أو من فرع تاني لشركتك وليكن أو جبت ال File دا من restricted Zone زي المواقع اللى بتستخدملها VPN عشان محظورة عليك وهكذا ...

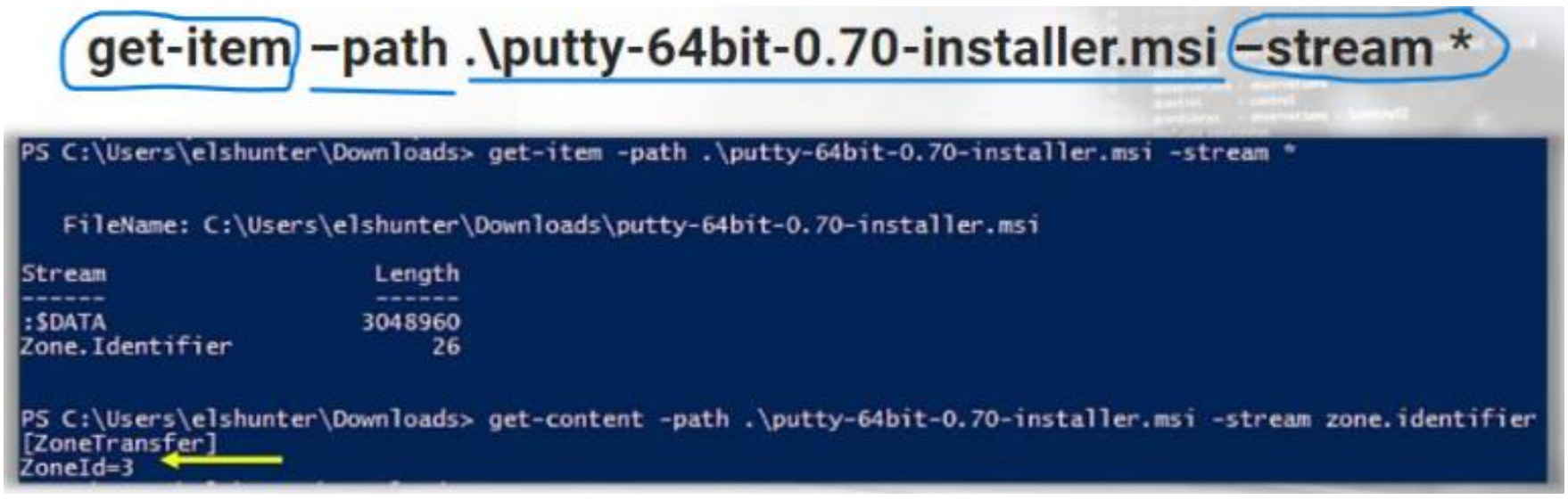
ـ علشان كدا اي ملف انت كUser بتحمله من الانترنت أو غيره زى مذكرنا بتقوك نسخه ال Windows اللى عندك واخده الملف دا وعاملاله Create ل ADS وجواه ال Zone parameter الخاص بيه ولكن مكان انت هتجيب منه الملف دا ال Zone number الخاص بيه .

Value	Zone
0	My Computer
1	Local Intranet Zone
2	Trusted Sites Zone
3	Internet Zone
4	Restricted Sites Zone

ـ فأي ملف بتحمله عن طريق برامج نظام **Windows** بيكون واخد **Zone identifier Number** يكون معرفه ودا اللى بيفسرلك لو انت منزل برنامج وجيت تشغله لقيت ال **Windows** مش راضي يشغلهولك فأنت تعرف علطول ان ال **Zone identifier** بتاعت الملف دا مش معروفه بالنسبه لل **Windows** أو مش مسموحلك اساسا انك تشغل ملف زي دا عشان انت جايبه من **Zone** ولتكن **Restricted** فأنت جايبه بطريقه مش مشروعه فمش مسموحلك تشغله .



- في **Command** عندنا من خلال ال **Power Shell** اللى عندك فال **Windows** تقدر تشوف بيه ال **Zone Identifier** بتاعت اي ملف انت نزلته وليكن هنا على سبيل المثال حملت ملف خاص ببرنامج ال **Putty** اللى بتستخدمه لما تكون عاوز تعمل **Connection** بال **FTP** أو **Telnet** ... وال **Command** هو **Get-content** وبعد كدا تديله اسم ال **Path** الخاص بالملف وبعد كدا اسم الملف وبتقوله **-** **stream** وبعدين **Zone. identifier** وهتلاقيه جابلك رقم ال **Zone** بالاضافه لل **Domain** اللى اتحمل منه وكمان ال **Path** اللى اتحمل منه الملف اللى انت شاكك فيه هيجبلك عنه كل حاجه .



ـ فأحنا عندنا ك Threat Hunters رقمين من ال Zone Identifier لازم تحطهم تحت سقف الشك وهم ال 3 و4 الاخصين بأي ملف تم تحميله من ال Internet وال Restricted zone اللى هو المكان مش مسموحلك تحمل منه اي ملفات يعني Un safe site فتاخد بالك حتى من ال 3 اللى هو جي من ال Internet يظل برضه Untrusted لحد منعمله ال Scan ونتأكد انه Clean .

ـ طب أفرض لقينا ملف معموله Download وال Zone Identifier الخاصه بيه ممسوحه أو مش موجوده فساعتها وانت مغمض اعرف ان ال Malware دخل عالكود الخاص بيه وعدل فيه بحيث انه يمسح ال Zone identifier بتعته علشان انت ك hunter لما تروح ب Command زي ال Get-content متعرفش تعرف ال Zone number بتاعت ال File اللى اتعمله Download فزي متقول ال Malware بيخفى الشبهه عنه فأنت أول متلاقي File بالشكل دا أعرف ان دي طريقه من طرق التخفي لل Malware فتشك فال File أكتر وتعمله Analysis وتتأكد منه ... زي المثال دا تماما الخاص بال Malware اللى اسمه Coin miner كان بيستخدم الطريقه دي فالتخفي .

- نيجي لتاني طريقه عندنا من طرق ال Evasion لل Malware وهي ال Injection بأنواعها ... ال Malwares بتحتاج انها تخبي نفسها جوا ال programs تانيه عالنظام عندك عشان تفضل شغاله من غير محد يكتشف دا فبتستخدم طرق مختلفه عشان تعمل لنفسها Inject أو للكود بتعها جوا Process معينه عندك عالنظام وتشتغل كأنها جزء منها ومثال على ذلك حقن مكتبات ال DLL وال Memory وال Process وغيرهم ... والكلام دا هنشوفه فالجي تفصيلي .

- تعالى نتكلم الاول عن ال DLL injection ودا من اشهر الطرق اللى بيستخدمها ال Malware عشان يدخل نفسه جوا ال Processes اللى شغاله حاليا عال System عندك .

- أول الخطوات لل Malware هي ال Locate Process اللى هي تحديد العمليه اللى فيها ال Malware هيعمل Inject لل DLL file بتاعه ... فلازم الاول نحدد ال Process بتعتنا ... الكلام دا بيتم عن طريق ال Windows API زي ال CreateToolhelp32Snapshot() ودي المسؤوله عن انها بتاخد Screen Shot لكل ال Processes الل شغاله على الجهاز بتاعك ... وعندك ال Process32first() ودي المسؤوله عن انها تجبلك أول Process فال Screen shot اللى اتاخدت عن طريق ال API اللى قبلها وعندك ال Process32next() دي المسؤوله عن تخليك تتنقل مبين ال Processes واحده ورا التانيه لحد متلاقي ال Target المناسب بتاعك ... فببساطه خالص ال Malware بتاعك بياخد Screen Shot لكل ال Processes اللى شغاله عندك وبيبدء يفحص ال Processes دي واحده واحده ولما يلاقي ال Process اللى عاوز يعمل فيها Inject هيكمل فالخطوات اللى هنشوفها مع بعض دي زي تخصيص Address فال memory وكمان نحمل فيها ملف ال DLL وكمان ننفذ الكود ال Malicious بتعنا جوا ال Process ويبان أكنه عادي .

- بعد معملنا تحديد لل Process اللى هيحقن فيها ال Malware ال DLL بتاعه فيها ... تعالى نعمل Open process اللى هو فتح العمليه اللى استهدفناها ودا عن طريق برضه ال Windows API زي ال Get module handle() ودي بتجيب ال Address المعين ل Module جوا ال Process ... وال Get proc address() ودي بتجيب عنوان ال Function جوا ال Module ودا مهم لل Malware لو هيستخدم Function موجوده فعلا ... وعندك ال Open process() ودي بتفتح ال Process المستهدفه عشان ال Malware يقدر يعدل عليها .

- الخطوه اللى بعد كدا هي ال Allocate memory اننا نخصص Address جوا ال memory عشان نحط فيها ال Malicious Code بتعنا أو نكتب فيه مسار ال DLL file اللى هنحقنه ودا بيتم عن طريق ال Virtual AllocEX() ودا ال API اللى بيستخدمه ال Malware عشان يحجز مكان فال Memory الخاصه بال Target Process بحيث زي مقولنا نقدر نكتب فيها ال Malicious code أو المسار الخاص بال DLL file بتعنا اللى عاوزين نحمله .

- يبقا احنا لحد هنا فالطريقه التانيه لل Malware Evasion اللى هي ال Injections وكل دا بنفهم الطريقه بتاعت شغل ال Malware واازي بيصيب جهازنا أو ال End point وبعد ذلك هنشوف ازاي نعمله Detect ... عملنا ايه عشان متوهش مني ... ال Malware حدد ال Process اللى هيشتغل عليها وفتحها عشان يقدر يعدل عليها ويتعامل معاها وحجز مساحه فال Memory الخاصه بال Process اللى ترجتها دي عشان يعمل Inject للكود بتاعه أو يحقن ال Path الخاص بال DLL عشان يتنفذ أكنه جزء طبيعي من ال Process .

- تعالى نروح للخطوتين المهمين اللى بيعملهم ال Malware عشان ينفذ ال DLL ال Malicious جوا ال Target Process .

- ان ال Malware هيعمل Copy لمسار ال DLL file جوا ال Process عشان ال Process تقدر توصله وتشغله ... ودا عن طريق ال Function ال Write process memory() ودي بتساعد ال Malware بتعنا يقدر يعمل Write لاي Data فال Memory Process اللى فتحناها قبل كدا وساعتها هيقوم كاتب ال Path الخاص بال Malicious DLL ... وتاني خطوه معانا وهي ال Execute لل DLL ... بعد مكتبنا ال Path بتاع ال DLL جوا ال memory تعالى نخلى ال Process بتعتنا تشغل ال DLL file ال Malicious على اساس انه عادي عالنظام ... ودا هيتم عن طريق اننا نعمل Createل Thread جديده جوا ال Target Process ودا عن طريق ال Create remote thread() ودي اللى بتخلى ال Malware عندنا يكريت Thread جديده جوا ال Target process ويقولها شغلى الكود الموجود فال Dll file ... وبعد كدا عن طريق ال Load Library() هنحمل ال Malicious DLL File بتعنا جوا ال Process ودا اللى فعليا هيخلى الكود بتعنا يشتغل .

- الخلاصه لحد دلوقتي ... ال malware حدد ال Process اللى هيعمل فيها Inject للكود وفتح ال Process اللى مترجتها لعمليه حقن ال DLL File وخد صلاحيات التعديل عليها وحجز مكان فال memory عشان يكتب فيها مسار ال DLL file وقام كاتب المسار جوا ال Memory وقام بعد كدا مشغل ال Malicious DLL file عن طريق انشاء Thread جديده جوا ال Process المستهدفه وبكدا ال Malicious DLL file اشتغل جوا ال Process لل APP اللى شغال وكأنه جزء منها بشكل طبيعي ويقدر ينفذ اي حاجه ال Malware عاوزها منه .

ـ فيه نقطه مهمه هنا وهي ان ال Create Remote Thread() دي مش الطريقه الوحيده اللى ممكن يستخدمها ال Malware لو عاوز يعمل Inject ل Malicious DLL file ف Process معينه عند ال Target ... نوضح النقطه دي .

ـ عندك بعض ال Documents APIs اللى شركه زي Microsoft بتشرجها وبتوضحها فشروحتها ان ال Malware ممكن يستغلها بالطريقه اللى ذكرناها فوق ... لكن عالجانب الأخرعندك ال Un Documents APIs ودي Functions موجوده فال Windows ولكن Microsoft مش بتوضحها او بتشرحها لل Developers .

ـ دا مهم لينا ك Threat Hunters عشان ال Attackers بيستخدموا ال Un Documents APIs الموجوده برضه عندك فال System عشان يتجنبوا ال Detect من ال Antivirus لأن خد بالك من نقطه برامج ال Antivirus بتكون مجهزة عشان تعمل Detect لل Functions المعروفه زي ال Create Remote Thread() لكن مش اشتراطا انها تعمل Monitor لكل Functions الغير معروفه زي Nt create Thread Ex() وال Rtl create user Thread() وال Functions دي أقل شهره وبتأدي نفس الغرض عادي فهي مش معروفه لل Antivirus زي ال Create Remote Thread() .

ـ تعالى نتكلم شويه عن ال Un Documents APIs اللى هي مش متعارفه زي ال Nt Create Thread Ex() بتسمحلك بأنشاء Thread جديد فأي Process ودي ممكن تحملها من Ntdll.dll علشان مش متاحه فال API العاديه لل Windows ودا عباره عن ملف نظام اساسي فال Windows اللى هو ال Ntdll.dll .

- وبعد كدا ال Rtl Create User Thread() نفس قصه اللى فاتت بتستخدمها لأنشاء Thread جديده جوا اي Process شغاله عندك واللى بيخليها خطيره ان عندك أدوات زي ال Mimi Katz اللى بنستخدمها عشان نجيب ال Password من ال Memory وكمان ال Metasploit الغنيه عن التعريف بيستخدموا ال Function دي عشان يعملوا Inject لل Malicious code بتعهم جوا ال Process ويتجنبوا ال Antivirus أو يعملوله Bypass .

-عندنا طريقتين من ال DLL Injection لازم نكون عارفنهم وهما ال DLL Classic وال DLL Reflective طب ايه الفرق بينهم ؟!

- ال Classic DLL Injection دي عشان تشتغل عند ال Target لازم يكون ملف ال DLL ال Malicious موجود على ال Disk اللى هو القرص الصلب اللى عند ال Target فجهازه ... فبيتحط ف Path معين على جهاز ال Target وبعد كدا بيتم تحمليه جوا ال Process المستهدفه عند ال Target ... عيبه ان ال Antivirus يقدر يكتشفه بسهوله لأنه بيكون ملف موجود عال Disk فعليا فيقدر يوصله .

- اما ال Reflective DLL دي أكتر ذكاء وتخفي عن ال Classic فبدل متخزن ملف ال DLL على ال Disk بتقوم محملاه بشكل مباشر فال Memory بدون أي تأثير على ال Hard Disk ومستوى التخفي هنا عالى جدا ودا بسبب ان برامج الحمايه زي ال Antivirus بتعتمد على فحص الملفات الموجوده على ال Disk وال Reflective ملهاش دعوه بال Disk أصلا فصعب أكتشافها من ال Antivirus .

- فال DLL Injection التقليدي اللى هو ال Classic لازم تستخدم ال Windows API زي ال Load library زي مقولنا قبل كدا عشان تحمل ال DLL file واللى ممكن برامج الحمايه تراقبه وتكتشف ال Attack دا ... أما فال reflective مفيش استخدام ل Windows API وبالتالي ال Attack بيكون مخفي أكتر .

- ال Malicious DLL بيشتغل ازاي فحاله ال Reflected DLL Injection ... أول حاجه انه مش بيحتاج Loader خارجي فال DLL نفسه بيقدر يحمل نفسه فال Memory ... كمان بيقدر يعمل Import لل Addresses لوحده !! يعني يقدر يستخدم ال Functions الخاصه بال Windows بدون منحتاج نحمله بالطريقه ال Classic ... وكمان بيقدر يعمل Relocations يعنى يعدل فال Addresses الموجوده فال Memory لو اتنقل من مكان لأخر ... يعني بالمختصر هتلاقي ملف ال DLL بيشتغل تلقاءي داخل أي Process عندك من غير مال Windows يحس بأي حاجه ودا اللى بيخلى ال Attackers يحبوا يستخدموا النوع دا من ال DLL injections .

- عندنا Tools جاهزة بتدعم ال feature اللى هي ال Reflected DLL injections زي ال Metasploit الغنيه عن التعريف وال PowerShell Empire دي Tool بنستخدموها فال Pen testing بأستخدام ال Power Shell .

- وطبعا ال APTs Groups مبيستخدموش ال Automated Tools دي لاء بيطوروا أدواتهم الخاصه بال C وال C+ وبيعملوا Inject لل DLL بأسلوب متطور ومختلف بحيث ميتمش كشفها من ال Antivirus بكل سهوله .

- تعالى نروح للطريقه التالته عندنا فال Malware Evasion اللى هي ال Thread Hijacking .

- دي عباره عن تقنيه متطورة بيقدر ال Malware يستخدمها عشان ي Inject الكود بتاعه من غير ميعمل Create ل Process أو Thread جديده ودا طبعا بيقلل من فرص أكتشافه ... الفكره ببساطه ان بدل مال Malware يشغل كود جديد بنفسه بيبحث عن الخيط أو ال Thread شغال على النظام حاليا ويخطفه ويدمج الكود ال Malicious بتاعه جواه ... بالضبط زي واحد بيسرق عربيه ماشيه بدل ميجيب عربيه جديده .

- أول الخطوات لازم ال Malware يلاقي ال Thread اللى هيحقن فيها الكود أو يعمل Locate لل Thread وفالحاله دي هيستخدم Windows API Functions زي CreateToolhelp32Snapshot() ودي بتخليك تاخد Screen shot لكل ال Threads الموجوده عندك عالنظام ... وال Thread32 first() بيجبلك أول Thread وال Thread32 next ودا بيقعد يقلب مبين باقي ال Threads عشان يجبلك ال Target المناسب .

- المميز فالطريقه دي اننا مبنشغلش Process جديده عال System عندنا فبنقلل احتماليه ان يتعملنا Detect وكمان بنستغل ال Threads الموجوده عال System وبالتالى ال resources بتاعت الجهاز استهلاكها هيقل وكمان زي مقولنا أصعب ان يتعملها Detect مقارنا بال Classic DLL ... فال Attacker بيحط كود ف Thread شغال أصلا عندك عال System عشان محدش يشك فيه .

- ال Malware بعد اما يحدد ال Thread المناسبه لازم يفتحها عشان عشان يعرف يتعامل معاها وبيستخدم فالحاله دي ال Function ال Open Thread() عشان يبقا ليه صلاحيه التعديل عليها ... وبعد كدا بيعمل Suspend لل Thread يعنى بيوقفها مؤقتا عشان ميحصلش اي مشكله وهنا بنستخدم ال Windows API اللى هي Suspend Thread() بيوقف تشغيل ال Thread مؤقتا عشان يسمح بحقن الكود بطريقه مناسبه ... لو عاوز تشبيه يقربلك الدنيا زي لما تمسك عربيه ماشيه أول حاجه بتعملها انك تفتح بابها وتوقفها عشان تقدر تتحكم فيها بالكامل بالضبط نفس القصه بتحصل هنا .

- بعد كدا ال Malware بيبدء يعمل Allocate لل memory يعني عاوز مكان فال memory عشان يخزن فيه ال Malicious code بتاعه اللى هو مثلا ال Shellcode أو ال path الخاص بال malicious DLL file ودا بيتم عن طريق ال Function اللى هي ال Virtual AllocEX() ودي بتخصص جزء من ال Memory الخاصه بال Injected process عشان نحط فيها ال malicious code ... وبعد كدا بنعمل Copy لل malicious code فالجزء اللى خصصه ال malware لنفسه فال memory ودا بيتم عن طريق ال Write process memory() دي بتعمل لل Malicious code زي ال DLL أو ال Shellcode جوا مكان ال Memory اللى خصصه لكدا .

- بعد كدا هنعمل Resume لل Thread بتعنا اللى كنا عملناله Suspend لو تفتكر ! دلوقتي بعد اما ال Malware عمل Inject لل Malicious code فال memory address واتعمله Copy زي موضحنا ... جه وقت اننا نشغل ال Thread المصاب عشان يبدء تنفيذ ال malicious code اللى اتعمله Inject جواه ... دا بيتم عن طريق ال Function اللى هي Resume Thread() وبكدا ال malicious code شغال كأنه جزء طبيعي من ال APP اللى شغال .

- الطريقه الرابعه معانا وهي ال PE Injection ودي اختصار ل Portable Executable Injection ... دا أسلوب لل Injection زي ال DLL لكن الفرق بينهم ان ملف ال Malicious DLL دا مش لازم يكون متخزن فال Hard ... بدل ميكتب ال Path بتاع ال DLL فال memory بيستخدم ال Function اللى هي Write process memory() عشان يكتب ال Malicious code نفسه جوا ال Memory الخاصه بال Target process زي مكنا قولنا قبل كدا ... وكمان مش بيستدعي ال Load Library() لأننا بالفعل عملنا Inject لل Malicious code جوا ال Target APP فال process أقصد .

- فال PE Injection بأختصار بيكتب الكود ال malicious بشكل مباشر فال Memory الخاصه بال APP وكمان مش بيسيب أي أثر على ال Hard ودا طبعا بيصعب في أكتشافه أو ان يتعمله Detect وكمان مش بيستخدم ال Load Library() لأنه مش بيحمل ملف خارجي ... الكود نفسه هو اللى بيتعمله Inject .

- لما ال Attacker يعمل لل PE ال Inject ف Process تاني فبيبقا ليه Base Address جديد فال Memory اللى هو ال PE بتعنا لأن ال Process اللى احنا مترجتنها هي اللى بتتحكم فمكان التحميل ... ال Base Address دا هو اللى بيبدء منه ال PE فال Memory وبيتغير لما بنعمله Inject جوا Process مختلفه ... المشكله اللى عندنا هي ان ال Address دا بيتغير والكود جوا ال PE مش هيلاقي ال Addresses الاصليه بتعته فلازم يعمل لنفسه تعديل عن طريق ال Relocation Descriptions ودي اللى هتساعده انه يلاقي ال Addresses الحقيقيه .

- الطريقه الخامسه معانا هي ال Process Hollowing ... الفكره هنا ان ال Malware بيفتح Process عادي بس بيكون حالته Suspended يعني متوقف مؤقت وبعد كدا بيعمل تفريغ للكود الأصلي وبيكتب مكانه الكود ال Malicious بتاعه وبكدا يظهر كأنه APP عادي لكنه فالحقيقه Malicious .

- نشوف ال Attack دا بيتم ازاي؟؟!... أول حاجه عن طريق ال Windows API Function اللى هي CreateProcess() دي ال Malware هيعمل Create ل New Process في وضع مؤقت زي مقولنا أو Suspended ودي اللى هنعمل فيها بعد كدا Host لل Malicious Code ... ثم بعد كدا هنعمل ال Un-map Memory اللى هو هنمسح الكود القديم ... ال Malware هيشغل برنامج عادي عندنا عال System زي ال Notepad وهيحطه فوضع مؤقت فلازم دلوقتي نمسح الكود الاصلي الخاص بال Notepad.exe عشان نحل محله بالكود ال Malicious بتعنا وهنا هنستخدم ال Windows API Function اللى هي ZwUnmapViewOfSection() أو ال NtUnmapViewOfSection() وال Functions دول وظيفتهم انهم يفرغوا الكود الاصلي لل Notepad.exe ... نيجي للخطوه اللى بعدها وهي ال Allocate & Write هنحجز مساحه لل Malicious Code وهنكتبه ... بعد أما مسحنا الكود الأصلي لل Notepad.exe لازم ال Malware يحجز Address فال Memory لل Malicious code ودا هيتم بنفس طريقه ال DLL Injection لو تفتكر ... عن طريق ال Functions اللى هي VirtualAllocEx() ودي بنستخدمها عشان نحجز Address في ال Memory بتاعت ال Process اللى ترجتناها اللى بتشغل ال Notepad.exe ... وال Function التانيه اللى هي WriteProcessMemory() ودي اللى هيكتب بيها ال Malware ال Malicious code بتاعه جوا ال Address الجديده اللى تم حجزناها .

- الخطوه اللى بعد كدا هي ال Set Entry Point وهي اننا نحدد نقطه الدخول ... بعد اما ال Malware بتعنا حط ال Malicious code جوا ال Process اللى ترجتناها عند ال Victim بنحتاج بعد كدا نغير نقطه البدايه اللى ال System بتاع ال Victim هيبدء منها ينفذ ال Instructions ... فالحاله دي هنستخدم ال Windows API Function اللى هي Set thread Context() ودي بتسمح بتعديل سياق التنفيذ اللى اتكلمنا عليه بحيث لما ال Process اللى ترجتناها تبدء التشغيل وتنفذ ال Instructions المطلوبه منها تنفذ ال Malicious code بدل الكود الأصلي لل Notepad.exe .

- نروح للخطوه الأخير بعد كدا وهي ال Resume اننا هنشغل ال Process اللى عند ال Victim اللى ترجتناها من جديد ... احنا كنا عاملين ليها Suspend لحد منحقن ال Malicious code بتعنا جواها وبعد كدا عملنا تغيير لنقطه الدخول ساعتها نطلع بقا ال Process بتعتنا من حاله التوقف اللى هي فيها ال Suspended State وبعد كدا ال malware هيبدء فتنفيذ ال Process عند ال Victim وطبعا ال System عند ال Victim هيفتكر ان كله تمام ودي Process عاديه شغاله عالنظام بس فالحقيقه هي جواها Malicious code بتاع ال Malware وبكدا ال Antivirus القديمه مش هتقدر تكتشف الكلام دا وهيبان أكنها Process عاديه شغاله عالنظام لكن الجديد بيخش جوا ال Process ويفتش عالكود فلو لقي ال Technique الخاص بال Process Hollowing هيعرف يعمله Detect .

- الطريقه السادسه عندنا هي ال Hook Injection ... ال Technique دا بيستخدمه ال Malware عشان يراقب أو يعمل Monitor لل Events اللى بتحصل عال System ...

- بمعنى ال Malware بيحط HOOK أو خطاف ال System وأول ميحصل Event معين زي ضغطه زرار أو حركه للماوس من ال User يبدأ ينفذ ال Malicious Commands بتعته اللى هيعملها Inject ... الكلام دا بيتم عن طريق Function من ال Windows API Functions وهي ال Set Windows Hook EX() دي بتسمح لل Malware انه يربط نفسه مع Events شغاله عالنظام حاليا ويعترضها قبل متوصل لل APP الفعلى اللى عتتنفذ فيه ويعدل فيها بال Malicious Code بتاعه .

- الكلام دا هتلاقيه بيستخدمه ال Malware فبعض الطرق زي فال Key logging زي انه يراقب ضغطات ال User على ال Keyboard ويبدء يسجل كل حاجه وبعد كدا ال Stealing Credential's انه يسرق ال بيانات تسجيل الدخول الخاصه ب User معين بمجرد مال User يكتبها وبعد كدا فحاجه زي ال UI Manipulation انه يتلاعب ويغير فواجهه ال User اللى هي ال User Interface ويخدعه عشان يدخل بيانات حساسه زي ال Visa Card Data وغيرها ... ودا انت مش هتحس بيه هتلاقيه شغال فصمت وبيسرق ال Data بتعتك عن طريق ال Keyboard وبيعمل Hook ل Sensitive Data وهتلاقيها مستخدمه كتير في برامج ال Spyware ... وعندك ال WH_Keyboard دي عشان نعمل Monitor للكيبورد والتانيه ال WH_Mouse عشان نعمل Monitor للماوس ... زي مقولنا اي User هيكتب حاجه على الكيبورد أو يحرك الماوس فالكود اللى زرع ال Hook هيقدر يشوف ال Event دا ويتفاعل معاه عادي ودا كله بيتم عن طريق ال Function اللى هي SetWindowsHookEX() زي مذكرنا ودي بتسمحلك انك تختار نوع ال Hook اللى عاوز تستخدمه واهمهم ال WH_Keyboard وال WH_Mouse لأننا ال Attackers هيستفادوا منه فحته التجسس وسرقه ال Sensitive Data .

- الطريقه السابعه من طرق ال Malware Evasion هي ال Kernel Mode Rootkits زي ال SSDT Hooks ... ايه دي!؟.

- ال SSDT دا اختصار ل System Service Descriptor Table ودي عباره عن جداول فال Kernel النواه الخاصه بال System اللى هو هنا Windows وليكن ... ال System بيستخدم الجداول دي عشان ال Functions الاساسيه للنظام موجوده فين ... زي ال Functions اللى بتتعامل مع ال Files وال Network وال Processes وغيره ... وكل Input فالجداول دي بيشير الى كود وظيفي معين في ملفات ال System الاساسيه ... خد مثال ... زي ال ntoskrnl.exe دا بيعتبر قلب نظام ال Windows لانه بيحتوى على كل الأشياء اللى محتاجها ال System عندك ... برضه ال Win32k.sys دا مسؤول عن الجرافيك والواجهه الرسوميه فال Windows ... تمام فهمنا ال SSTD بتعمل ايه ... تعالى نشوف موضوع ال SSDT Hooking .

- ال SSDT Hooking ببساطه ان ال Attacker بيلعبوا فالجدول دا وبدل متخلى ال SSDT يشاور على ال Functions الأصليه بيخليه يشاور على دوال تانيه تنفذ حاجات Malicious ... الكلام دا هتلاقيه في نوع من ال Malwares اسمه ال Rootkit كنا اتكلمنا عليه فالأول ارجعله ... ال Root kits بتستغل التقنيه دي عشان تتحكم فال System ومحدش يقدر يكشفها ... فلو مثلا عندك Virus عاوز يستخبى فيعدل على Function بتتحكم في عرض الملفات بحيث لو ال Windows حب يجيب ملفات الفيروس دا ... ال Function اللى عدلناها تمنع ظهوره ... وبرضه ال Antivirus بيستخدمه بنفس الفكره ولكن لأغراض حميده فبدل مالفيروسات تعدل على الجدول فهو اللى بيسبقهم ويعمل Monitor لل System ومكوناته عشان يكشف اي نشاط مشبوه عليه قبل ال Attacker .

- الموضوع دا شغال ازاي ... تعالى نشوف مع بعض التفاصيل .

- اي برنامج عندنا فوضع ال User Mode زى اي برنامج عادي بتشغله على Windows فيجي البرنامج دا يطلب اي Service من ال Kernel بيبعت الطلب عن طريق ال Native API وبعدين الطلب بيروح لل SSDT عشان يعرف ال Kernel المفروض يشغل انهو Function بالضبط ... عندك برنامج في وضع ال User بيطلب تنفيذ حاجه من ال Kernel زى فتح ملف مثلا ... الطلب دا بيدخل على ال SSDT عشان يعرف انهو Function فال Kernel مسؤوله عن المهمه دي اللى هي فتح الملف ... وبعد كدا ال Kernel بينفذ المطلوب منه عن طريق ال Function المناسبه ... هنا ال Attacker بيتدخل ويروح زي مقولنا يعدل فال SSDT Table عن طريق ال Pointers الموجوده جوا ال SSDT فبدل مالطلب يروح لل Function الأصليه يروح ل Function تانيه معدله من ال Attacker وهتنفذ فالغالب Malicious Code ... ال Function المسؤوله عن العمليه دي هي ال Key Service Descriptor table () ودي موجوده جوا ال Kernel بتاع ال Windows وظيفتها انها توفر معلومات عن ال SSDT وال SSDT بفكرك الجدول بتعنا اللى بيحدد كل ال Services وال Functions اللى ال Kernel بيوفرها لل User Mode .

- الخطوره فال Technique دا انه بيأثرعال System بالكامل مش مجرد برنامج واحد بمعني اي حد عال System هيحاول يستخدم ال Services الموجوده عال System هيتم أعتراضه من ال Malicious Code كأنك بالضبط بتغير عنوان Google Map لعنوانك فكل اللى كان رايح للمكان الأصلي الأول هيجيلك عالعنوان الجديد.

- ال SSDT Hooking بيتم عن طريق الخطوات التاليه وهي ان أول حاجه Hook SSDT بمعنى ال Attacker بيقوم مدخل Entry معين فال SSDT واللى بيكون مسؤول عن تنفيذ Function معينه زي ال NTQueryDirectoryFile() وال Function دي مسؤوله عن انها تجبلك قايمه ال Files من اي Folder فال System ... بس طبعا ال Attacker بيدخل بدالها ال Malicious Input عشان لما نستدعي ال Function دي بعد كدا نستدعي ال Malicious code بدالها .

- بعد كدا ال Attacker بيعمل ال Call Function ... بعد أما عدلنا ال Input فال SSDT ... أي برنامج هيطلب تشغيل ال Function الأصليه NTQueryDirectoryFile() مش هتشتغل ال Function الأصليه لاء هتشتغل ال Function اللى كتبها ال Attacker بدل الأصليه واللى هتستدعي ال Malicious Code ... هي هذا السياق هتلاقي ال Attacker بيكتب Code يخلى ال Function دي متعرضش ملفات معينه زي ملفات ال Virus أو ال Rootkit مثلا ... وكمان يتجسس على الملفات اللى بيفتحها ال User أو بيستخدمها عال System وتبعتها لل Attacker برضه عن طريق ال Code اللى هيعدله ال Attacker جوا ال Function الأصليه عندك ال Victim عال System وكل دا عن طريق ال SSDT زي موضحنا من خلال التلاعب بيها .

- الخطوة اللى بعد كدا وهي ال Pass Control بمعني بعد اما ال Malicious Function اللى ذكرناها هتتنفذ هنروح بعد كدا نستدعي ال Function الاصليه بتاعت ال Windows زي ال NTQuery Directory File عشان تجيب ال Data اللى ممكن تعرضها زي مثلا بعض ال Files في Folder معين وهكذا ... الخطوه اللى بعد كدا والأخيره هي ال Alter & Return Results بمعنى هنا ال Rootkit بيتلاعب ويغير فال results اللى جياله ...

فلو فيه Malicious file أو Folder ما عاوز يخفيه فيعدل فال Data دي ويرجع لل APP اللى كان طالب ال Data دي والملف ال Malicious كأنه مش موجود أصلا ... فالفكره هنا ال Rootkit بيستخبى جوا ال System بطريقه ذكيه تمنع اكتشافه فأول ميحصل استدعاء ل Function معينه تلاقيه بيعدل فال Results بتعتها قبل متوصل لل User وعن طريق ال Technique دا يقدر يخفي اي Files أو Processes أو اي حاجه تانيه عن ال User وال Antivirus الخاص بيه .

- نيجي للطريقه التامنه معانا وهي ال IRPs Hooks فال Kernel mode Rootkits ... يعني ايه ال IRPs فالأول ؟ ... اختصار ال (I/O Request Packet) دي ببساطه زي رساله أو Request بيستخدمها ال Operating System عشان ينقل ال Data بين المكونات المختلفه لل System زي مثلا لما APP معين يطلب أنه ي Write أو Read على Hard Disk عندك فال PC ... فلو مثلا ال Antivirus بيحاول يقرء Files معينه فال Rootkit ممكن يعدل على ال Data اللى راجعه ويخلى ال Antivirus يفتكر ان ال Files دي مش موجوده أصلا والهدف من دا اخفاء ال Malicious files من ال Antivirus ودا بيتم عن طريق تعديل وتلاعب ال Rootkit فال Data اللى مبين مكونات ال System ...

- كل حاجه جوا ال Windows Kernel معتمده على ال IRPs سواء كنا فال Network ال (TCP, UDP) أو نظام ال File System أو ال Mouse أو ال Keyboard وكمان ال Drivers اللى شغاله عال System وعلشان كدا اي حد يقدر يتلاعب بال IRPs يقدر يسيطر على حاجات كتير جوا ال Windows بدون ميتم أكتشافه بسهوله من ال Defensive Tools اللى عند ال User ... بص كدا المثال دا .

```
DriverObject->MajorFunction[IRP_MJ_CREATE]=DiskPerfCreate;
DriverObject->MajorFunction[IRP_MJ_READ]=DiskPerfReadWrite;
DriverObject->MajorFunction[IRP_MJ_WRITE]=DiskPerfReadWrite;
DriverObject->MajorFunction[IRP_MJ_SYSTEM_CONTROL]=DiskPerfWmi;
DriverObject->MajorFunction[IRP_MJ_SHUTDOWN]=DiskPerfShutdownFlush;
DriverObject->MajorFunction[IRP_MJ_FLUSH_BUFFERS]=DiskPerfShutdownFlush;
DriverObject->MajorFunction[IRP_MJ_PNP]=DiskPerfDispatchPnp;
DriverObject->MajorFunction[IRP_MJ_POWER]=DiskPerfDispatchPower;
```

- دا جزء من كود تابع لل Microsoft WinDDK ودا اختصار ال Windows Driver Development Kit وبيوضح ازاي ال IRPs بتكون متصله بوظايف معينه فال Drivers زي ...

IRP_MJ_CREATE ودا مسؤول عن انشاء ال Files وكمان ال IRP_MJ_READ ودا مسؤول عن قرايه ال data وكمان ال IRP_MJ_WRITE ودا مسؤول عن ال Write فال Files وكمان ال IRP_MJ_SHUTDOWN ودا مسؤول عن ال Shutdown لل System ... وهتلاقي غيرهم من الوظايف المهمه فأداره ال System ال IRPs هتلاقيه داخل فيها فدي توضحلك أهميته فال System ... ليه مهم تاخد بالك من الكلام دا ! لأن ال Rootkit هتلاقيه يقدر ياخد Control على واحده من ال IRPs دول ويقدر يسيطر عال Processes المرتبطه بيها بشكل كامل زي المثال اللى قولنا عليه فوق انه يقدر يمنع ال Antivirus من انه يمسح بعض ال Files أو يخلى ال System بتاعك يعتقد انه فيه File معين مش موجود وفالحقيقه هو شغال فال Background .

- أي جهاز أو Driver عندنا فال Operating System ليه جدول وظايف اللى هو ال Function Table ودا دوره انه يحدد ازاي الجهاز بتاعك بيتعامل مع ال Requests اللى بتوصله ... ال Attackers بيستخدموا تقنيه اسمها ال DKOM اختصارا ل Direct Kernel Object Manipulation علشان يعدلوا على ال Pointers اللى هي المؤشرات اللى فالجدول ودا بيساعدهم بيغيروا ال Behavior بتاع ال System بدون مال User يحس ... خليني ابسطها .

ـ تخيل عندك قايمه بتقول لكل Request فال System عندنا هيتنفذ ازاي ... فلو حد عرف يتلاعب بالطلبات دي يقدر يوجهها لغرض تاني غير الغرض اللى المفروض تأديه ... دا بالضبط اللى ال Attackers عاوزينه من ال DKOM بمعني هيستخدموا التقنيه دي في انهم يعملوا Hidden لل Processes ال malicious فتلاقي ال Rootkit شغال عند ال User عال System وهو لايعلم شيء لأن ال DKOM تقدر تتلاعب بال kernel بشكل مباشر ودا اللى بيخليها فمنتهي الخطورة لو ال Attacker استغلها لأنها بتديه تحكم على أجزاء حساسه من ال system ... وال Kernel هو العقل المدبر والمتحكم فال System عندك مشؤول عن اداره ال Processes وال التحكم فال Drivers واداره ال Memory وغيره من وظايف ال System واي Process شغاله عال System ليها Object فال Kernel وال DKOM يقدر يعدل فال Object دا يخليه ينفذ حاجه تانيه أو يخفي ملفات Malicious معينه خاصه بال Rootkit ... فمثلا تلاقي ال Attacker يستخدم ال DKOM فأنه يخفي ال Rootkit من ال Task manager عن طريق انه يمسح ال Process بتعته من ال Process List وبكدا هتختفي من ال Task Manger وال Antivirus مش هيشوفها .

ـ تعالى نشوف مثال عملي لل IRPs عشان الدنيا توضح أكتر .

```
old_power_irp=DriverObject->MajorFunction[IRP_MJ_POWER];
DriverObject->MajorFunction[IRP_MJ_POWER]=my_new_irp;
```

ـ هنا ال Attacker هيعدل على ال Pointers بتاعت ال Function الخاصه بال IRPs Requests بحيث يتم تبديل ال Function الأصليه الخاصه بال System ب Function تكون Malicious ال Attacker عاوز يوجهك ليها .

ـ المثال اللى قدامك دا فيه ال Hooking بيتم على خطوتين ... الأولى اننا نحفظ ال Function الأصليه في Variable أسمه التالي...
old_power_irp وبيتم تخزين ال Data الاصليه اللى بتتعامل مع بتتعامل مع ال Power Requests ... عن طريق الكود دا .

```
old_power_irp = DriverObject->MajorFunction[IRP_MJ_POWER];
```

ـ الخطوه التانيه ان ال Attacker هيستبدلها ب Function تانيه عن طريق الكود دا ...

```
DriverObject->MajorFunction[IRP_MJ_POWER] = my_new_irp;
```

ـ هنا غيرنا ال Function الأصليه وحطينا واحده جديده ودا معناه ان اي Power Request جديد هيتم التعامل معاه بال Function الجديده ال Malicious بدل الأصليه ... وال Power Request هنا المقصود بيها وضع الجهاز بتاعك عامله Sleep أو Shutdown .

ـ خد بالك من نقطه ... ال Request بتاع ال IRPs لما بيجي يتنفذ ال System مش بينفذه علطول أومال بيحصل ايه !؟... ال Request دا بيعدي على كذا طبقه فال System أو كذا مستوى يعني وكل طبقه منهم بتشوف ال request وتعمل معاه شغلها قبل ميوصل للمرحله الأخيره بمعنى ... عندك مرحله ال Pre-Processing دا أول مال request يوصل من ال Hardware لل System فبيقوم ال System شايف ال Request ويفلتره ويشوف هل هو صالح أو Malicious ويقدر يعدل فيه كمان ويقبله أو يرفضه ... المرحله اللى بعد كدا وهي ال Post-Processing ودي فيها ال Request بعد اما يخلص لف على كل الطبقات اللى تحته فالطبقه دي بتتأكد انه اتنفذ بشكل صحيح ولو فيه خطأ او شكله مش مضبوط بيتم التعديل عليه أو رفضه ودا كله عشان نضمن ان ال Requests بتعدي على مراحل معينه عشان مفيش Errors تحصل لل Data أو ال Hardware .

- فال Attacker بيستغل الكلام دا عشان يعدل فال request دي ويبدلها بال Malicious Function اللى تخلى مثلا الجهاز بتاعك في وضع السكون أو ميعملش Shutdown بحيث ال Rootkit يفضل شغال عليه وخد عندك أستفدات كتير لا تحصى من حوار ال IRPs دا .

- الطريقه التاسعه معانا وهي ال Userland Rootkits زي ال IAT Hooks ... ال IAT هي ال Import Address Table ودا ببساطه عباره عن جدول بيحتفظ بعناويين ال Functions اللى أي APP محتاجها من مكتبات الDLL عشان يشتغل ... الفكره ان اي APP فال Windows هتلاقيه مبيحتويش على كل الأكواد اللى بيحتاجها لكنه بيعتمد على مكتبات خارجيه اللى هي ال DLLs زي اللى كنا ذكرناهم اللى هما ال user32.dll وال Kernel32.dll وغيرهم ... فال IAT بيكون فيه قايمه بكل ال Functions اللى APP هيستخدمها من ال DLLs مع ال Location بتعها فال Memory ... دا هيفد ال APP فأيه !؟؟ أول حاجه ال APP ميضيعش وقت يقعد يدور على ال Functions فبيلاقيها جاهزة فال IAT وتاني شيء هيخلي ال APP يشتغل بكفاءه بدل ميقعد يحمل أكواد زياده مش هيستخدمها فدا بيخفف عليه ... ال Attackers يقدروا يستغلوا الكلام دا فأنهم يغيروا ال Addresses بتاعت ال Functions دي ويبدلوها ب Addresses تانيه خاصه بيه فبدل مال APP ينادي على Function معينه لاء هينادي على ال Function اللى ال Attacker بدلها مكان الأصليه .

- نروح للنوع العاشر عندنا وهو من نفس فصيل ال Userland Rootkit ولكن مختلف فالنوع وهو ال EAT Hooks ودا اختصار ل Export Address Table اللى هو جدول ال Addresses اللى بيتم تصديره من ال DLLs ... طب سؤال الفرق بينه وبين ال IAT فأيه؟... ال IAT كان الجدول اللى بيستخدمه ال APP عشان يعرف ال Functions اللى بيشتغل بيها .

ـ لكن ال EAT الموضوع جوا ملفات ال DLL نفسها وبيشتغل على ملفات ال DLL فقط لأنه التعديل أو التلاعب من ال Attacker بيتم على ال Addresses الأصليه لل Functions جوا ملف ال DLL نفسه ... بمعني أي APP بيستخدم ال Function من ال DLL هيتم تحويله لل Function ال Malicious بسبب التعديل فال EAT ... فال EAT ملوش تأثير على ال EXE file ولكن شغله كله على ال DLL file ودا عكس ال IAT تماما لأنه بيشتغل على التلاعب فالجدول اللى ال APPs بتستخدمه عشان تعمل Import ل Functions معينه من ال DLLs ودا معناه ان التعديل بيتم على مستوى ال APP اللى هو EXE أو ال DLL اللى بيستخدمها ... وبكدا نكون عرفنا الفرق مبين ال IAT وال EAT ... ودا اللى يفسر ان ال Attackers تركيزهم بيبقا على ال EAT أكتر من ال IAT لأن بعض ال Antivirus بتراقب ال IAT وبتغض النظر عن ال EAT فبيكون التلاعب أكتر فال EAT لما يجي يعمل Bypassing Security لل Defensive Devices اللى عند ال User .

ـ الطريقه ال 11 معانا هو ال Userland Rootkits هو ال Inline Hooks ... دا أخطر نوع من ال Hooking والفكره هنا ان ال Attacker مش هيعدل فجدول أو Pointers زي ال EAT,IAT هتلاقيه بيعدل جوا الكود نفسه بتاع ال Function اللى هي API Function ... بمعنى أخر بدل مال APP ينفذ الكود الأصلي لل Function هتلاقي ال Attacker بيحط ال Malicious code جوا ال Function أول متيجي تنفذ الكود اللى جواها هتنفذ ال Malicious بدل الكود الأصلي .

ـ هنا ال Attacker بيبص عالكود الأصلي بتاع ال Function اللى عاوز يخترقها وبياخد أول شويه Bytes من الكود اللى هما أول Bytes لما الداله بتشتغل ويعدلهم وبيحط بدالهم كود جديد...

واللى بيكون ال Jump Instruction عشان يوجه التنفيذ لمكان تاني فال Memory اللى فيه ال Malicious code ودا هيتم عن طريق ال Instruction Point اللى هو ال EIP ودا المؤشر اللى بيحدد الكود اللى عليه التنفيذ فال Attacker بيروح لل Malicious code بدل الكود الأصلي ... نفس فكره ال Buffer Overflow Attack مع الفوارق طبعا .

ـ خد مثال عشان تفهم ... لو عندك APP عاوز يطبع جمله زي printf("Hello World!"); فلو حد أستخدم Inline Hooking على Function زي Printf ممكن يخليها تعمل Print بدل اللى فوق يطبع كدا ... printf("You've been hacked!"); ودا من غير ميتلاعب فالكود الأصلي بتاعك بس من خلال التلاعب بال Function .

ـ الطريقه ال 12 معانا وهي ال Process Hiding من ال Rootkits ... الفكره هنا ان ال Rootkit بيحاول يخبي نفسه جوا ال System عشان محدش يعمله Detect فلو عندك فال Soc ال Hunter أو Analyst او حتى برامج الحمايه زي ال Antivirus عال PCs بيدور على ال Malicious Code اللى شغال هتلاقي ال Rootkit بيخبي ال Process اللى شغال منها كأنه مش موجود... فلو فتحت ال Task Manager عشان تشوف ال Processes اللى شغاله عندك عال System مش هتلاقي ال Process بتاعت ال Rootkit شغاله لأنه عملها Hide وشغال فال Background .

ـ ال Rootkit مش بيخبي نفسه فقط لاء دا بيغير حاجات فال System عشان يخلي ال Malicious Process تكون مخفيه تماما ... ودا بيحصل عن طريق ال Native API Function اللى هي NtOpenProcess ودي بنستخدمها فال Windows عشان نفتح Process ونشوف ال Data بتعتها ونتحكم فيها ونعدل فيها .

- ودا بيتم عن طريق ال SSDT Hooking اللى كنا شلرحناه بالتفصيل فوق ... اللى هو ال System Service Descriptor table ودا الجدول الموجود فال Kernel بيحتوى على كل ال Addresses الخاصه بال Functions المهمه عال System واللى بيستخدمها ال System دايما ... ال Rootkit بيستخدم ال SSDT عشان يغير ال Address بتاع ال Function اللى هي NtOpenProcess فالجدول فبدل ميخليه يروح يشاور على ال Function الأصليه لاء هيروح لل Function المزيفه اللى عاملها Create ال Rootkit اساسا وبكدا ال Rootkit يكون عمل ال Hooking على ال Function اللى هي NtOpenProcess وبكدا اي APP يحاول يشوف ال Processes اللي شغاله هتلاقي ال Rootkit بيقوله " ارجع مفيش حاجه هنا " وبكدا APP زي ال Task manager مش هيعرف يشوف ال Malicious Process الل شغاله .

- بعد كدا عاوزين نخفي ال Process ال Malicious اللى شغاله عن طريق ال EPROCESS ودا عباره عن Data Structure هيكل بيانات بيحتوي على جميع المعلومات الخاصه ببال Processes اللي شغاله وال System عندنا بيستخدم ال EPROCESS عشان يتابع كل ال Processes اللى شغاله عليه فال Rootkit بتعنا بيشيل ال Process ال Malicious بتعته من ال EPROCESS وبكدا أي APP زي ال Task manager حاول يشوف ال Process اللي شغاله حتى من ال Kernel مش هيلاقيها ... جوا ال EPROCESS فيه حاجه مهمه عندنا وهي ال Active process links ودي زي سلسله بتربط ال Processes ببعضها لأن كل Process عندنا ليها ال Forward link وكمان ال Backward link عشان تعرف مين ال Process اللى قبلها ومين اللى بعدها زي نظام الدايره كدا كل Process مربوطه بلى جنبها ... ال Rootkit بتعنا عاوز يخبي نفسه من الدايره دي يعمل ايه !؟؟

ـ فبيقوم رايح فاكك ارتباط ال Malicious Processes دي من ال Active process links زي موضحنا ازاي ؟ ... بيقوم ماسك ال Forward Link وال backward Link اللى بتوصل ال Process دي ويخليهم يعملوا Bypass لل Malicious Process بتعته وبكدا أي حد يبص على دايره ال Processes مش هيشوف ال Malicious Process بتاعت ال Rootkit ... ولو ال Rootkit عباره عن Driver متحمل جوا ال Kernel فعندك List تانيه ال System بيتابع بيها ال Drivers المحمله واسمها ال PsLoadedModuleList فال Rootkit مش هيكتفي انه يخفي نفسه من ال Processes فقط لاء كمان هيشيل نفسه من ال PsLoadedModuleList فكدا حتى لو حد حب يدور على ال Drivers اللى شغاله مش هيلاقيها وبكدا حتى لو ال Rootkit كان Driver فعمل Hidden نفسه .

ـ الطريقه ال 13 معانا وهي ال Masquerading ... دي بنقصد بيها التنكر بمعنى... دا اللى بيعمله ال Malware عشان محدش يعمله Detect بيتنكر وبيمثل انه حد تاني أو شخص تاني مش المتنكر هوضحلك الدنيا ... بدل مال Attacker يتعب نفسه ويستخدم تقنيات معقده عشان يخفي ال Malware فال Malware بدل ميسمي نفسه مثلا Virus.exe لاء دا بيسمي نفسه على اسم APP شغال عندك عال System بالفعل زي Chrome فتلاقي ال Attacker منكر ال Malware دا بنفس الاسم زي Chrome.exe ودا عشان ال Antivirus ميشكش فيه ويعديه .

ـ فال Malware هتلاقيه بيسمي نفسه بنفس أسم ملفات موجوده عال System زي مثلا ... svch0st حط بدل حرف ال O ال Zero ... وبرضه scvhost هتلاقيه هنا بدل الحرفين اللى بعد حرف ال S ... وبرضه svchost32 هتلاقيه بيضيف رقم .

ـ وكل الأمثله دي قريبه من الملف الأصلي الموجود عال System اسمه svchost.exe ودا اللى بيكون شغال دايما فال Background لل Windows فلو حد فتح ال Task manager ولقى دا شغال مش هيشك فيه ... الكلام دا بالأضافه ان ال Malware هتلاقيه بيحط نفسه ف أماكن حساسه عندك فال System زي C:\Windows\ وال C:\Windows\System32\ ودول Folders فال System عندك فيها ملفات مهمه فلما ال Malware يزرع نفسه هناك هيبان كأنه ملف شرعي أو legitimate عال System زي أي APP أساسي فال Windows ودا هيصعب من طرق أكتشافه ... بل وكمان ال Malware بيخفي نفسه ف Temporary folders وزي ال Temporary Internet files ودا بيخزن ملفات الانترنت اللى بتتصفحها انت ك User زي ال Cache وال Cookies ودا مكان مناسب عشان يستخبى فيه ال Malware كأنه ملف انترنت مؤقت ... وكمان بيستخبى فال Program files زي برامج ال Windows' الأصليه فلو حد دخل على ال Folder هيلاقيه مليان برامج وملفات شرعيه ودا بيصعب من أكتشاف ال Malware ... ودا طبعا مش كل طرق التخفي لل Malware عن طريق ال Masquerading لاء ال Malware هتلاقيه بيتفنن فالتخفي فممكن تلاقيه في سله المهملات اللى هي ال Recycle bin أو اي مكان فال System يلاقيه مناسب وكمان بيدور على مكان فال System أنظمه الحمايه أو ال Antivirus مبيبصش فيه بسهوله فلازم تاخد بالك وتعمل Investigation كويس جدا عندك عال System فالأماكن الحساسه.

ـ الطريقه ال 14 معانا من طرق ال Malware Evasion وهي ال Compression & packing بمعنى ... فالأول ايه هو ال packer ؟؟ دا عباره عن برنامج وظيفته انه يضغط الملفات التنفيذيه اللى هي ال executable Files (exe file) والهدف منه كان تصغير حجم الملف بدل ميكون حجمه مثلا 50 ميجا يبقا 10 ميجا ...

عشان يبقا أسهل فالنقل والتحميل مبين الأجهزة ... علاقه ال Malware بالموضوع دا لما ال Attacker بيجي يكتب ال Malware بيستخدم ال Packer مش بس عشان يصغر حجم الملف لكن كمان عشان يخفي ال Malware من ال Antivirus !! طب دا بيحصل ازاي ؟! تعالى نشوف ...

- برامج الحمايه أو ال Antivirus بتدور على نمط أو بصمه جوا ملفات معينه عال System لل Malware ودا زي توقيع رقمي لل Malware يميزه ويعرفنا ك Threat Hunters ان دا هو فعلا ال Malware حاجه زي ال IOCs بالضبط كدا ... فلما الملف يتضغط عن طريق ال Packer النمط أو التوقيع دا بيتغير أو يختفى بشكل مؤقت مع عمليه الضغط دي ... فال Attacker بيستخدم ال Packer فالحته دي ولما بيتم ضغط ملف خاص بال Malware ال Signature بتاعه بيتغير أو بيختفي مؤقتا ولذلك ال Antivirus مبيعرفش يتعرف عليه أو يعمله Detect ... وكلما حجم الملف دا قل كلما عدد ال Signatures اللى برامج الحمايه ممكن تعملها Detect تقل ودا طبعا بيقلل من فرص ان ال Malware يتعمله Detect من ال Antivirus .

- ال Attackers المحترفين هتلاقيهم مش بيروحوا يستخدموا برامج ال Packers الموجوده جاهزة لاء ... دول بيروحوا يصمموا برامج من الصفر خاصه بيهم ... وعندك مثال زي Yoda packer دا واحد من ال Packers اللى اتعمل مخصوص عشان يصعب فرص أكتشاف ال Malware ويخفيه ... وعندك الناس التقليديه اللى مبيحبوش يتعبوا نفسهم ف تصميم Packer جديد خاص بيهم هتلاقيهم بيستخدموا أدوات مشهوره زي UPX اللى هي Ultimate packer for executables وال UPX معمول عشان يقلل حجم الملفات التنفيذيه ال exe يعني بشكل شرعي عال System ولكن ال Attackers بيستخدموه عشان يشفروا ال malware عال System ويخفوه .

ـ وخد بالك من نقطه وهي ان ال UPX لو استخدمه Attacker ما بشكل صحيح فعندك برامج الحمايه المتقدمه دلوقتي ممكن تفك الضغط اللى عمله ال Packer للملف وتفحص الملف وساعتها هتعمله Detect ولكن لو Attacker ذكي شويه بيعمل عليه بعض التعديلات أو أستخدم Packer مصممه بنفسه زي مقولنا ساعتها ال Antivirus بيوقف عاجز ومبيعرفش يتصرف فخد بالك ك Threat Hunter عشان بعض ال Attackers متقدمين فالجزءيه دي وفالأغلب الهجمات الغير تقليديه هتلاقي ال Attackers بيصنعوا ال Tools بتعتهم بنفسهم .

ـ الطريقه ال 15 معانا هي ال Recompiling ... هنا نقصد اعاده التجميع !؟ بمعنى ... أي ملف تنفيذي عندنا exe file بيكون ليه بصمه أو Signature وبنسميها ال Hash زي ال MD5 Hash مثلا ... ال Hash دا برامج الحمايه بتستخدمه عشان تتعرف على ال Malware فلو ال Hash معروف انه Malicious هتلاقي برامج الحمايه بتحذفه علطول وعندك خير مثال على ال Check على ال Hash هو Virus total ... الموقع دا بتديله Hash ويقولك اذا كان دا Malicious ولالاء عن طريق ال Signatures اللى محتفظ بيها عنده فال Database بيقارنها بال Signature بتاعت ال Hash اللى عطتهوله ويرد عليك .

ـ ال Attackers بيعملوا ال Recompiling لل Malware عشان يهربوا من القصه اللى فوق دي ... ال attacker بيقوم واخد نفس الكود لملف ال exe وبيعيد برجمته وترجمته وفكه وتجميعه عن طريق Compiler ولكن ببعض التغيرات البسيطه فالكود ودا لأن كل مره بيتم تجميع الكود بيتم توليد Hash جديد حتى لو الكود هوهو برضه مدام اتفكك وجمعته تاني بدون تعديل حتى هتلاقي اتعمل Generate ل Hash جديد تاني غير الأول مع أنه نفس الكود متعدلش عليه ومغيرتش فيه حاجه !!.

ـ ونتيجه للكلام اللى فات دا هتلاقي برامج الحمايه مش عارفه تعمل Detect لل Malware لأن ال Hash كل شويه بيتغير وطبعا المواقع دي بتحدث نفسها أول بأول بأخر ال Signature بس تظل معندهاش القدره على تجميع كل ال Hashes الخاصه بال Malwares عشان حته التعديل اللى بتم من خلال ال Recompiling اللى وضحناها دي ... فال Malware لو تم التعديل عليه من خلال Compiler فال Hash أتغير وبالتالى برامج الحمايه مش هتلاقي ال Hash الجديد فال databases بتعتها وبالتالي مش هتعرف تعمل لل Malware ال Detection .

ـ الطريقه ال 16 معانا وهي ال Obfuscation اللى هي التعتيم أو التشويش ودي طريقه بيستخدمها ال Malware Development عشان يصعبوا على ال Analysts انهم يفهموا الكود عن طريق انهم بيعدلوا الكود بطريقه تخليه صعب الفهم أو التحليل بس فنفس الوقت يفضل شغال زى مهو من غير ميتأثر ... بمعني الكود بيبقا مغلف أو مشوش علشان لو حد من ال Analysts حب يعمل Reverse Engineering هندسه عكسيه للكود ويفككه بيبقا صعب ولكن مش مستحيل ... البرامج الشرعيه عندك عال System برضه بتستخدمه زيها زي البرامج الخبيثه ولكن الفارق ان البرمجيات الخبيثه بتستخدمه عشان تهرب من برامج الحمايه وتصعب جدا ان يتعملها اكتشاف من ال Antivirus ... انما البرامج الشرعيه بتستخدم الطريقه دي عشان تحمي نفسها من سرقه الكود بتعها وان يتعمله Reverse Engineering ... والبرامج الخبيثه اللى بيعملها Create ال Malware Development كل متعمل Infect لجهاز فال Network تروح تغير فالكود بتعها كل شويه عشان يبقا ليه Signature وشكل جديد عال System ودا بيصعب على برامج الحمايه زي ال Antivirus مثلا انه يكتشفها.

- نيجي للطريقه ال 17 والأخيره من طريق ال Malware Evasion وهي ال Anti-Reversing Technique's ... ودي ببساطه التقنيات المضاده للهندسه العكسيه (Reverse Engineering) ودول عباره عن بعض الطرق اللى بيستخدمها ال Malware عشان يصعب على Analysts فهم الكود بتاعه ال Malicious وتحليله والهدف منه ان ال Attackers يلغبطوا ال Analyst ويضيعوا وقته عشان ميقدرش يوصل لأزاي ال Malware بيشتغل ويفهم من خلاله تفكيكه للكود ال behavior بتاعه .

- عندنا بعض الطرق اللى بيستخدمها ال Malware عشان يلغبط بيها ال Analysts فال SOC وأولهم هو ال Virtual Machine ... ال Analysts دايما هتلاقيهم بيعزلوا ال Malware وبيستخدموا فكدا ال Virtual machine عشان تكون الدنيا أمان فممكن ال Malware يبقا فيه كود ما يعرفه انه شغال ف Virtual machine فيوقف نفسه أو يغير ال Behavior بتاعه ويشتغل بشكل مختلف... تاني حاجه معانا وهي ال Debugger ودي Tool بيستخدموها ال Analysts عشان يتابعوا الكود خطوة بخطوه فممكن ال Malware يعمل Detect لل Tool دي ولو لقاها هيبدء يتصرف ب Behavior مختلف حتى لو حكمت انه يقفل نفسه ويوقف شغله عشان ال Analyst ميقدرش يستوعب ويفهم الدنيا ماشيه ازاي ... الطريقه التالته عندنا وهي ال Junk Data وهي ان ال Malware أحيانا بيضيف أكواد ملهاش لازمه ولا وظيفه حقيقيه غير انها تلغبط وتشتت ال Analyst والأكواد دي بتخلي شكل الكود معقد ومليان تفاصيل ملهاش لازمه زي موضحنا ودا كله برضه عشان يصعب ال Reverse Engineering على ال Analyst اللى هيفك الكود بتاعه ويحلله فبيحاول يصعبها عليه ويكسب وقت هو فشغله ... وبكدا نكون أنهينا الحديث عن ال Malware Evasion Techniques وطبعا زي موضحنا فيه طرق كتير تانيه وكل يوم فيه جديد فلازم تكون انت Updated بكل شيء.

5.Malware Persistence:

- هنا كلامنا هيكون عن ال Persistence لل Malware وهي الأستمراريه ... بمعني أخر ازاي ال Malware تفضل موجوده عند ال Victim PC حتى لو عمل restart أو يحاول يعملها Delete بمعني تاني ازاي ال Malware تقدر تحافظ على وجودها وتستمر عند ال Victim ... عندك دليل بيشرح كل الطرق اللى بيستخدموها ال Attackers فال Malware اللى بيترجتوا بيه ال Victim عشان ال Malware يحقق الأهداف المطلوبه منه عند ال Victim ودا هو ال MITRE's & ATTACK ودا لازم ال Threat Hunter يكون Updated بيه علطول لأنه زي مقولنا بيشرح كل الطرق اللى بيستخدمها ال Attackers وهتلاقيه ليه Website موجود عال Internet عادي تقدر توصله وتستعين بيه .

- أول طريقه من ال Persistence هي ال AutoStart Locations بمعنى الأماكن اللى ال Malware بتعنا بيخبي نفسه فيها علشان يشتغل تلقاءي كل مره تفتح الجهاز بتاعك ك Victim ... الفكره هنا ببساطه ان ال malware عشان يفضل شغال حتى لو ال User عمل restart للجهاز بتلاقيه يحط نفسه ف أماكن معينه فال Windows بتخلى ال Windows يشغله أتوماتيك مجرد مال user ي Login لجهازه ... وأشهر مكان ال Malware بيستخدمه هو ال Registry والمسار اللى بتكلم عليه جوا ال Registry أهو هتلاقي ال Malware بيشغل نفسه من المسار دا...

HKLM\Software\Microsoft\Windows\CurrentVersion\Run

- الMalware بيضيف نفسه فالمكان دا ولما ال User يفتح جهازه ويعمل Login فال Windows يقرء الموجود فال Path دا ويشغل ال Malware من غير ميحس ال User بحاجه ... ودي من الطرق المنتشره عشان ال Malware يثبت نفسه على جهاز ال Victim .

- هنا هنذكر الأماكن اللى ال Malware بيستخبى فيهم ويشتغل منهم بشكل تلقاءي اللي هي ال AutoStart Locations .

- أول مكان عندنا هو ال

HKCU\Software\Microsoft\Windows\CurrentVersion\Run

- بالمناسبه ال HKCU يعني HKEY_CURRENT_USER بمعنى دا مرتبط بال User اللى عندنا عالجهاز دلوقتي ... فال Malware لو حط نفسه هنا هيشتغل كل مره ال User يعمل فيها Login .

- وتاني مكان معانا هو ال

HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

- وال HKLM بمعنى HKEY_LOCAL_MACHINE ودا مرتبط بكل ال Users عالأجهزة عندك ... والمكان دا بيخلى ال Malware يشتغل لكل User يعمل Login .

- تالت مكان معانا هو ال

HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

- دا زي ال Path الأول بالضبط ولكن دا خاص بال ’Policies اللى بيستخدمها ال Windows للسياسات وال Setting الخاصه بال User ... وبرضه ال Malware هنا هيشتغل مع ال Login لل User الحالي عال System ... فال Malware هتلاقيه دايما بيدور على أكتر من مكان عشان يضمن انه يشتغل بشكل تلقاءي بعد كل Restart أو Login وبيختار دايما أماكن مش واضحه لل User العادي عشان ميتعملوش Detect بالساهل .

- فيه بعض ال Locations أقل شيوعا مش مشهورة أوي اللى ال Malware بيستخدمها عشان يستخبى فيها ويشتغل منها بشكل تلقاءي.

-أول مكان معانا هو ال HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit

- المكان دا خاص بال Processes اللى بتشتغل مع تسجيل الدخول ال Login ... فلو ال Malware دخل هنا فالمكان دا هيشتغل أول مال User يعمل Login قبل حتى ميظهرعال Desktop .

تاني مكان معانا هو ال HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options

- المكان دا خاص بال Developers عشان يقدروا يعملوا Debug لل APPs فال Malware هتلاقيه بيحط نفسه فال Location دا عشان يتنفذ بدل APPs تانيه موجوده فنفس ال Location دا .

- المكان التالت معانا هو ال HKLM\Software\Wow6432Node\Windows NT\CurrentVersion\Image File Execution Options

- نفس فكره اللى فات ولكن خاص بال APPs ال 32 bit على ال Operating System ال 64 bit ... فهنا ال malware هتلاقيه بيشتغل ال APPs القديمه ال 32 bit علشان يشتغل من غير ميتعمله Detect ... فالأماكن دي مش واضحه لل User العادي ودا بيدي فرصه لل Malware يستخبى فيها وينفذ نفسه بسهوله ... وخد بالك انت ك Threat Hunter الأماكن دي مهمه بالنسبالك وانت بتعمل Hunt عشان لو فيه Malware شغال عندك عال System وخافي نفسه وانت متعرفش وبيشتغل بشكل Automatic هتلاقيه فالأماكن دي وفيه أماكن أخري أكيد انت المفروض تبحث عنها وتبقا عارفها برضه ال Malware بيترجتها عشان يستخبي فيها ويعمل Persistence لنفسه عال Target machine عشان يقدر يشغل نفسه فحاله انت عملت Shutdown لجهازك فأول متعمل start هتلاقيه أشتغل معاك مجرد متعمل Login لنسخه ال Windows بتعتك وبكدا يبقا حقق عنصر الأستمراريه عند ال Victim .

ـ عندك tool مهمه متستغناش عنها ك Threat Hunter وهي ال Autoruns من حزمه ال Sys internals دي بتوريك أول بأول الحاجات وال APPs اللى بتشتغل أول مال Windows يفتح وبتعرضلك الأماكن المختلفه جوا ال Registry وال Folders اللى ممكن ال Malware يستخدمها ودي بتساعدك فال Hunt ... ولو حابب تتعمق وتعرف الطرق الجديده اللى بيستخدمها ال Malware عشان يعمل ال Persistence Technique ممكن تتابع ال MITRE'S & ATTACK دي عباره عن موقع اساسي لل Threat hunters بيقدروا من خلاله يتعرفوا على ال أخر ال Technique's بتاعت ال Attackers فال Attacks فهتلاقي فيها ما يخص طرق ال Attackers فأنهم يخفوا ال Malware بأحدث التقنيات والطرق .

ـ تاني طريقه معانا من ال Persistence هي ال Scheduled Task ... بمعنى عندنا أدوات فال Windows زي ال at.exe و schtasks.exe الأدوات دي بتسمحلك تعمل جدوله لأي برنامج أو سكريبت عشان يشتغل فوقت معين أو لما يحصل Event معين زي مثلا لما ال PC يفتح أو يدخل User جديد عال System ... دا الكلام ال Normal انما ال Attacker بيستخدم الكلام دا أو الأدوات دي اللى ذكرناها عشان يشغل ال malware أول مالجهاز يفتح في وقت معين من غير مال User ياخد باله من حاجه ... وال Malware ممكن يشغل Scripts تانيه عشان يعمل Persistence أو يعمل Privilege Escalation مثلا ياخد صلاحيات ال Admin عالجهاز بدل مكان User عادي ...

ـ فلو لقيت Scheduled task مش متأكد منها أو مش عارف مصدرها فخلى بالك منها فدي من طرق شغل ال Malware عشان يفضل Persistence حتى لو عملت Shutdown أو Restart لجهازك ... تعالى نشوف مثال عملي .

- عندنا جروب APT معروف اسمه APT3 ومستخدمين الأمر دا ...

schtasks /create /tn "mysc" /tr C:\Users\Public\test.exe /sc ONLOGON /ru "System

- تعالى نفصص ال Command دا ونفهمه ... schtasks
/create ودا الأمر اللى بنستخدمه عشان ننشيء Task مجدوله جديده
... وعن طريق ال /tn "mysc" واسم ال Task الجديده هي
"mysc" وبعد كدا ال malware هيشتغل من خلال المسار دا /tr
C:\Users\Public\test.exe وفحالتنا الملف اللى هنشغله اللى بيمثل ال
Malware هو test.exe ... وبعد كدا /sc ONLOGON دي ال
task اللى هتشتغل أول مال User يعمل Login ... وبعد كدا /ru
"System" وال task دي هتشتغل بأعلى صلاحيات عندنا فال
System ودي أعلى صلاحيات فال System عندنا .

- الطريقه التالته معانا فال Persistence وهي ال COM
Hijacking ... دا أختصار ل Component Object Model ودا
عباره عن نظام من Microsoft عشان يخلى ال APPs تتكلم مع
بعضها فال System ... دا زي مترجم بيخلى APPs تشتغل مع بعضها
حتى لو اتكتبت بلغات برمجه مختلفه عن بعضها... ومن الحاجات اللى
بيتعمد عليها ال COM هي (Object Linking and Embedding) OLE
دا اللى بيخليك مثلا تحط جداول ال Excel جوا ملف Word وال
Active X دي أضافات بتشتغل جوا ال Browsers أو ال Apps
عشان تزود وظايف معينه ...

- ازاي ال Attacker ممكن يتلاعب بال COM ويستغله ... ال
Attacker ممكن يتلاعب بال COM عشان يخلى ال APPs تشغل ال
Malware بدل الكود الطبيعي ... فال Attacker بيغير ال Setting
فلما تفتح APP معين وال APP دا يشغل الكود ال Malicious بتاعه
بدل الكود الأصلي ... ودا بيساعد ال Attackers يخبى ال Malware
بتاعه ويخليه يشتغل عند ال target عادي من غير مال User ياخد
باله ... وبص عال MITRE'S & ATTACK هيفيدك أكتر فالحته دي .

- ال Attacker أو ال APT Groups بينفذوا ال Attack دا ازاي ؟!

- ال Hijacking دا بتعتمد على فكره أستبدال مكونات ال COM الأصليه بكود Malicious فال Attacker بيروح يغير حاجه اسمها ال References المراجع ... اللى هو المكون الأصلي لل COM ويحط بداله ال Malicious Code زي مقولنا فلما ال APP يحاول يشغل المكون دا بدل ميشغل النسخه السليمه هيشغل ال Malicious Code اللى زرعه ال Attacker بداله ... وزي مقولنا لو عاوز تفهم الطرق ال Advanced من ال Attack دا وأحدث الطرق شوفه من خلال ال MITRE'S & ATTACK الموقع دا هيفيد فحاجات كتير تابعه .

- الطريقه الرابعه معانا فال Persistence لل Malware هي ال (Search order) DLL Hijacking ... أي APP عندنا فال Windows عشان يشتغل وبيحتاج يستخدم مكتبه ال DLL بيروح يدور عليها ف أماكن بترتيب معين ودا المقصود من ال Search Order ... ال Attacker بيستغل الترتيب دا ويخدع ال App وهنشوف ازاي دا بيحصل تفصيلي بس خليك معايا ... فلو ال APP بيدور على مكتبه معينه وملهاش Path واضح فهيبدء يدورعليها ف Directory معينه بالترتيب !! ... تعالى نشوف العمليه بتحصل ازاي .

- ال APP بتعنا بيدور على ال DLL فنفس المجلد اللى هو موجود فيه اللى هو ال Local Directory اللى واقف فيه ال APP حاليا ولو ملقاش بيدور فمجلدات ال System فال path دا C:\Windows\System32 ... الخطوره هنا ان ال Attacker بيستغل دا فأنه يحط ملف Malicious DLL بنفس اسم المكتبه اللى ال APP بيدور عليها فنفس مجلد ال APP ... لأن ال APP بيدور فال Local Directory الأول هيلاقي ال Malicious file ويشغله على أساس انه الملف الصح وبكدا ال Malicious Code اتنفذ بسهوله ... وميزه النوع دا من ال Attack ان ممكن يتنفذ ب user عادي مش شرط صلاحيات عاليه .

- الطريقه الخامسه معانا وهي من ال DLL Hijacking ولكن نوع مختلف ... وهي ال Phantom DLL ... هنا ال Attacker بيستغل خاصيه موجوده فنظام ال Windows قديمه شويه وهي ان أحيانا ال APPs القديمه بتحاول تحمل مكتبات ال DLL القديمه واللى اغلبها بيكون ملغي من ال System الجديد أو مش موجود أصلا ... فال APP ساعات بيكون لسه بيدور عالمكتبات اللى اتشالت واتحذفت أو مبقتش مستخدمه ... ال Attacker بيقوم حاططلك ملف ال Malicious DLL بنفس أسم المكتبه اللى ال APP بتعنا بيدور عليها اللى هي قديمه ومش موجوده أو قد تكون أتحذفت ... بمعني ال Attacker عمل Phantom يعني شبح للمكتبه القديمه اللى اتحذفت وخدع ال System ...

- فال Attacker بيقوم مستبدل الملف الأصلي بتاع ال DLL بتاع APP معين بنسخه ملف Malicious بنفس الأسم بالضبط ... فلما ال APP يجي يشتغل ويدور على ملف ال DLL هيلاقي النسخه الجديه اللى هي Malicious وهيشغلها عادي بدل النسخه الأصليه... ال Attacker بيحدد المكان اللى ال APP بتعنا بيخزن فيه ملفات ال DLL بتعته ... فبيروح هناك ويعمل زي مقولنا يحذف الملف الأصلي ويحط مكانه الملف ال Malicious التاني وال APP بتعنا هيشغله عادي كأنه الملف الأصلي ولكنه فالحقيقه بيشغل ال Malicious code .

- ندخل عالطريقه السادسه معانا من ال persistence برضه من أنواع ال DLL Hijacking وهي ال Side Loading ... الفكره هنا ان ال Attacker بيستغل Folder اسمه WinSxS اختصارا ل Windows side-by-side ... ال Folder دا موجود فال Windows ووظيفته يساعد ال APPs انها تتعامل مع ال Problems اللى بتقابلها زي تكرار أو تحديث ملفات ال DLL .

- ال Attacker بيستغل ال Folder دا ويحط جواه DLL File بس بيكون Malicious بنفس أسم ملف DLL أصلي موجود عال System وال System هيشغله عادي ... ولما يجي ال APP يحمل الملف الأصلي اللى محتاجه هيحمل التاني ال Malicious اللى بيشيل اسمه برضه .

- الطريقه السابعه والأخيره من طرق ال Persistence لل Malware هي ال Windows Services ... وبتم على كذا مرحله وبشكل مختلف فنمشيهم بالترتيب ونفصصهم ... ال Service Creation ودا من أشهر الطرق اللى بيستخدمها ال malware عشان يعمل ال Persistence عند ال Victim Machine ودا عن طريق ال Command اللى اسمه Sc اختصار ال Service control ودا بيسمحلك انك بأنشاء وتعديل وحذف ال Processes عال System عندك ... فال Attacker بيروح يعمل Create ل Service جديده ويخليها تشغل Malicious file مع كل بدايه تشغيل للجهاز قبل حتى مال Antivirus يقوم ويكتشف الكلام دا ودا بيخلينا نعمل زي Bypass لل Antivirus ولكون مش بالطرق المباشره وهتفضل شغاله برضه حتى لو ال User عمل Restart للجهاز وميزتها انها هتوه وسط ال ألاف ال processes الشرعيه اللى شغاله بالفعل على نظامك فهي هتوه وسطهم وهيبقا صعب تطلع ال Malicious من وسط كل دول فهي هتعرف تختفي بشكل كويس .

دا مثال لو عاوز تعمل Create ل Service عشان الدنيا توضح أكتر.

```
sc create MaliciousService binPath= “C:\malware\evil.exe" start= auto”
```

- ال sc create عشان نعمل Service جديده واسمها ال malicious service وبعد كدا من خلال ال binpath بتديها المسار اللى فيه ال Malicious File بتاعك وبعد كدا ال start=auto يعني ال Service اللى عملنالها Create ...

هتشتغل بشكل أتوماتيك مع تشغيل ال System عندنا .

ـ ندخل على المرحله اللى بعدها فال Windows Services وهي ال Service Replacement ... وهنا ال Attackers مش بيعملوا create ل New Process من جديد لاء ... دول بيدوروا على Service شغاله عال System بالفعل ويبدلوها ب Malicious Service ... طب ازاي بيتم الكلام دا ؟ ... ال Attacker بيدور على Service ضعيفه فال System يعني ال service موجوده ولكن بيقدر يعدل عليها فال Setting الخاصه بيها بسبب الصلاحيات الضعيفه اللى عليها زي مثلا انها متسابه ل User عادي ممكن يشغلها أو يعدل عليها !! ... وبعد كدا ال Attacker بيقوم مستبدل ال EXE File بتاع ال Service دي بال Malicious file بتاعه فكل مال Service دي تشتغل هتشغل ال Malicious file معاها طبيعي ... وبعد كدا ال Attacker بيضبط من خلال ال Setting ال service دي انها تشتغل بشكل Automatic مع بدايه تشغيل جهاز ال Victim تشتغل ال Service دي معاه ... وعلى سبيل المثال توصيل المعلومه خد المثال دا ثبت بيها الدنيا .

```
sc config VulnerableService binPath= “C:\malware\evil.exe”
```

ـ ندخل عالمرحله الأخير معانا فال Windows Service وهي ال Service Recovery ... هنا ال attackers بيستغلوا فكره ال Service Recovery الموجوده فال Windows ... ودي وظيفتها انها تعيد تشغيل أي Service عندك عال System فحاله حصل فيها عطل أو وقفت فجأه ودي حاجه طبيعيه عشان تضمن أستقرار ال System ... لحد هنا كله تمام ... ال Attacker بيستغل الكلام دا فأنه يدخل لل Service ويعدل فيها ويضبطها انها لو وقفت أو حصلها عطل فال System يشغل Malicious file بدالها !! ...

ـ فبدل مال Service تشتغل من تاني بشكل طبيعي لاء هتشتغل ال malicious file اللى مجهزها ال Attacker بدالها ... وكل مره يحصل فيها عطل فال Service فال System مش هيعمل ال recovery لاء هيروح يشغل ال Malicious file بتاع ال Attacker اللى حددها فال Recovery فأي service تروح تعمل Recovery بنفسها هتلاقي الملف بتاع ال Attacker هناك فتنفذه ... فعلى سبيل المثال وتوصيل المعلومه لو عندنا Service أسمها My service وليكن فال Attacker ممكن يعدل فال Recovery Setting بتعتها ويخليها تشغل ملف اسمه Malware.exe فكل مره ال Service تقع فيها عن طريق ال Command التالي ...

```
sc failure My Service command= “C:\malware\malware.exe”
```

وبكدا ال Attacker يكون ضمن تشغيل ال malicious software بتاعه حتى لو ال Service دي وقعت أو حصلها مشكله ... بكدا نكون أنهينا حديثنا عن المراحل التلاته لل Windows Services وكمان نكون أنهينا حديثنا عن ال Malware Persistence Technique’s وذكرنا المشهور منها وفيه جديد كل يوم لازم تكون updated انت ك Threat Hunter بالكلام دا وتبلغ بيه مؤسستك.

ـ ختاما تعالى نلم اللى الكلام اللى شرحناه خلال ال Module دا فكام جمله... اتكلمنا فال Module دا عن تلت محاور أساسيه وهما ... أول حاجه اتعرفنا على أنواع ال Malwares وذكرناهم وتاني حاجه عرفنا ازاي ال Malware ممكن يعمل Infection لل System وبعد كدا دخلنا على تالت حاجه وهي ال Malware Evasion وهي التقنيات اللى بيستخدمها ال Malware عشان يهرب من ال detection ويعمل تخفي لنفسه عال System وبعد كدا أخيرا شوفنا ازاي ال Malware بيحاول يثبت نفسه فال system ازاي عن طريق ال technique’s الخاصه بال Persistence بأنواعها ... الكلام دا هينفعنا فال Module الجي واحنا بنعمل Malware Hunting .