Post-exploitation Techniques & Defense

Created By: Mohamed Ayman Elshelly

Privilege Escalation

Windows Privilege Escalation

- Stored Credentials
  - Common locations to find such files are:
    - C:\sysprep\sysprep.xml
    - C:\sysprep\sysprep.inf
    - C:\sysprep.inf
    - C:\unattend.xml
    - C:\Windows\Panther\Unattend.xml
    - C:\Windows\Panther\Unattend\Unattend.xml
  - Group Policy Preferences (GPP) inside SYSVOL
  - Detection
    - 1. Creating fake/honey files containing fake credentials and deploying them to the aforementioned locations.

      2. Then, we can monitor access to these files by first enabling file system auditing and then looking at any generated 4663 event related to these files.
    - 4663(S): An attempt was made to access an object.

      4625(F): An account failed to log on.

      4776(S, F): The computer attempted to validate the credentials for an account.
      - 4663
4625
4776
- Insufficiently Secure Service Registry Permissions
  - Local service configuration information are stored in the Windows registry under:

    - HKLM\SYSTEM\CurrentControlSet\Services.
    - Detection
      - Looking for Sysmon Event ID 1 entries that have a

        CommandLine field that contains something like reg add

        HKLM\SYSTEM\CurrentContr01Set\Services\XYZ / v ImagePath / d "path_to_a malicious executable. exe"

        and an IntegrityLevel field that contains something other than High.
      - Monitoring Sysmon's Event ID 13:

        RegistryEvent (Value Set)
      - Sysmon Event ID 1

        Sysmon Event ID 13
  - Insufficiently Secure Service Permissions

    Attackers may also have the ability to tamper with a service's binPath, if the service has been configured with lax permissions.

    If this is the case, attackers will try to introduce their own executable (which will be executed with the service's privileges), via the SC command.
    - Detection
      - Sysmon Event ID 1 entries that have a

        CommandLine field that contains something like

        sc config "service _ name" binPath= "path to_a suspicious executable .exe"

        or sc start "service _ name" and

        an IntegrityLevel field that contains something other than High.
      - Sysmon Event ID 1
- Unquoted Service Path
  - When configuring a Windows service, we should be careful to enclose the executable path in quotes. If we don't do so, when this service is starting Windows will try to locate and execute the executable inside every folder of the specified path until the executable is reached.

    "C:\Program Files\ATITechnologies\ATI.ACE\Fuel\Fuel.Service.exe"
  - Detection
    - by checking for Sysmon Event ID 1 entries where
      ParentImage is
      C:\ Windows\System32\services.exe and the

      CommandLine's beginning (in quotes) doesn't end with an extension and is the same as the Image path minus the extension.

      In addition the
      CommandLine field should also contain the remaining part of the path at the end, right after
    - Sysmon Event ID 1
- Insufficiently Protected Service Binary
  - Attackers may have the right to directly replace a service's executable, due to an insufficiently secure configuration.
  - Detection
    - Sysmon Event ID 1. Specifically, you will see a non privileged process (IntegrityLevel other than High)

      dropping an executable into a service's Image path (you should be aware of those paths)

      and this executable being executed with SYSTEM privileges (you will see that in a subsequent Event ID 1 entry).
    - Sysmon Event ID 1
- Always Install Elevated
  - AlwaysInstallElevated is policy that allows for the installation of a Microsoft Windows Installer Package (MSI) with system privileges, by a unprivileged user.
  - Detection
    - Sysmon Event ID 1 Specifically, you will see a non-privileged process (IntegrityLevel other than High) trying to quietly install a remote MSI

      (CommandLine msiexec.exe /q / I http://domain or address/ filename.msi).

      You will also notice an unprivileged user in the User field.
    - Then, in a subsequent (very close in terms of time) Event ID 1 entry that has

      C:\ Windows\System32\msiexec.exe

      specified in the ParentImage field, you will see a MSI being installed with SYSTEM privileges (IntegrityLevel System).

      You will also notice NT Authority\SYSTEM in the User field.
    - Sysmon Event ID 1

      EventID 1
    - Checking for Parent - Child process anomalies.
    - Specifically, you will most probably see a Sysmon Event ID 1 entry that is related to a privileged process (IntegrityLevel System) that has a ParentImage and ParentCommandLine of (Windows Installer-related) and a CommandLine field that contains cmd.exe or powershell.exe.
- Exploiting the Windows Kernel and 3rd-party Drivers for Privilege Escalation
  - CVE-2018-8120, which was related to a vulnerability discovered inside the Microsoft Windows Kernel 'Win32k.sys'.
  - Detection
    - We can detect the exploitation of kernel mode (or third-party driver) vulnerabilities for privilege escalation purposes by looking for medium integrity level processes that started with a non-SYSTEM token but spawned a child process with SYSTEM-level access.
- Abusing Windows Privileges for Privilege Escalation
  - Specific Windows privileges can be abused by attackers for privilege escalation purposes.

    Such privileges are:
    - SeDebugPrivilege
    - SeImpersonatePrivilege
    - SeAssignPrimaryPrivilege
    - SeTakeOwnershipPrivilege
    - SeRestorePrivilege
    - SeBackupPrivilege
    - SeLoadDriver
    - Token Privilege
    - SeTcbPrivilege
    - Detection
      - Sysmon has Event ID 8 to detect a big percentage of code injection attacks.
  - SeDebugPrivilege
    - Abuses the SeDebugPrivilege to inject malicious code into the winlogon.exe process, that is always running with SYSTEM level privileges.

      Injection was performed through the CreateRemoteThread function.

      This will allow him to execute commands with SYSTEM-level privileges
    - Detection
      - Sysmon Event ID 8

        - Find the included SourceProcessGuid in previous Sysmon Event ID 1

        - Find the included TargetProcessGuid in previous Sysmon Event ID 1

        - Sysmon Event ID 1 entries where Parent - Child anomalies are obvious.

Linux Privilege Escalation

- Look inside a Linux endpoint's command-line history

  Memory dump of a Linux endpoint can also provide you with the commands that were executed.
  - Kernel Version: uname -a
    - Operating System: cat / etc/ issue
    - Running Processes: ps auxw
    - Network Routes: route -n
    - DNS Server: cat /etc/resolv. conf
    - Arp Cache: arp -a
    - Current Network Connections: netstat —auntp
    - Current user permissions: find /user username
    - UID and GID Information for all users:
    - Last logged on users: last -a
    - Root accounts
    - Service Accounts: cat /etc/passwd
    - Home Directories: ls -las /home/*
    - current user execute: sudo -l
    - shadow file: cat /etc/shadow
  - Listen to a specific port: nc -l -p 1234
    - How your environment is configured and what your current shell is : env
    - Identifying the partition or "file" defined as the swap file : swapon -s
    - more..
  - Detection
    - Command History

Credential Theft & Cracking or Reuse

Windows Authentication Weaknesses

- LM/NTLMv1
  - The authentication protocol used between Windows clients and servers is called NTLM (NT LAN Manager). Although NTLM has been replaced by Kerberos, it is still widely used and supported in Windows machines. For example, it is used either when the client is authenticating to a server using an IP address or, when the client is authenticating to a server that does not belong to the same domain.
  - Detection
    - Wireshark: ntlmssp.ntlmserverchallenge and see if the challenge contain random values or not
- NTLMv2
  - The NTLMv2, introduced in Windows NT 4.0, still uses NT hashes, but with a much improved protocol.

SMB Relay

- SMB Relay attacks allow attackers to re-use authentication attempts in order to gain access to a system in the network.
- Responder & Inveigh
  - Performs SMB relaying but captures authentication attempts through LLMNR and NBT-NS spoofing/poisoning.

    - LLMNR (Link-Local Multicast Name Resolution) and NBT-NS (NetBIOS Name Service) spoofing are also two very effective methods for capturing users' NTLMv1 , NTLMv2 or LM (Lan Manager) hashes through a man-in the-middle type of attack.

    - Both LLMNR and NBT-NS allow for machines within a Windows-based network to find one another and are essentially a "fall-back" protocol used for the resolution of hostnames within a network when resolving of hostnames via DNS fails.

    - Attackers use the Responder and Inveigh tools for performing LLMNR and NBT-NS spoofing/poisoning, capturing NTLMv1 /v2 hashes and relaying them for authentication to other systems.
  - Detection
    - Sysmon logs (Event ID 3) for any SMB (or NetBIOS)-related communications with an untrusted IP (not a trusted File Server or Domain Controller), you will be able to identify if an SMB capturing infrastructure is operating inside the network.
      - Sysmon logs (Event ID 3)

        Port 443

        Destination IP Address
    - Using PowerShell script (you can name it as whatever_name.psl) and use GetWinEventData.psl
    - Responder (or Inveigh) Detection:
      1. send honey credentials fake and see the user that use this fake credentials.

      2. PowerShell and the CredDefense suite can assist you in that.

      3. PowerShell's script block logging capability. It can be utilized to detect Inveigh being loaded into the memory of an intranet machine.
      - Event ID 4648 - Fake Credentials

        ResponderGuard (part of CredDefense)

        Event ID 4014 - Execute a Remote Command

Pass the Hash

- if the attacker obtain a user's NTLM hash is perform an attack called "pass the hash", that can grant them access on a target, by means of the hash without using the actual plain-text password.

  Attackers performed pass the hash attacks through Metasploit psexec module.

  Tools: Metasploit psexec module

  Pass the hash is essentially lateral movement over the NTLM network. This means that NTLM connections are involved in the process.
- Specifically:
  1. It copies a binary to the ADMIN$ share over SMB
  2. It creates a service on the remote machine pointing to the abovementioned binary
  3. It remotely starts the service
  4. It stops the service and deletes the binary on exit
- Detection
  - Event ID 7045 and Windows Security Log Event ID 4697 can help in identifying new services.
    - Event ID 7045
Event ID 4697
  - When an NTLM connection occurs, Event ID 4624 with Logon Type 3 and Logon Process NtLmSsp is created on the targeted endpoint.

    Check the NTLM connection:
    4768 — A Kerberos authentication ticket (TGT) was requested
    4769 — A Kerberos service ticket (TGS) was requested
    4648 — A logon was attempted using explicit credentials
    4624 — An account was successfully logged on

    Logon types: 2 (Interactive), 7 (Unlock), 10 (RemoteInteractive) or 11 (CachedInteractive)
    - 4768
4769
4648
4624
  - Privileged NTLM connections can be identified by creating a correlation between the NTLM connection and event ID 4672. Event ID 4672 is related to a privileged account logging on a machine.
- Unfortunately for defenders, attackers have stepped up their game and now prefer mounting pass the hash attacks through WMI. The advantages of this technique is that no new service is ever created and no suspicious command is logged.
  - Detection
    - To be able to detect a pass the hash attack through WMI, logging for WMI events should be enabled. It is not enabled by default, this is why no suspicious command is logged during such attacks.

Pass the Ticket

- During a pass the ticket attack, the attacker extracts a Kerberos Ticket Granting Ticket (TGT) from a system's LSASS memory and then imports it on another system. The newly imported ticket can then be used to request Kerberos service tickets (TGS) and subsequently gain access to network resources.
- Detection
  - Pass the ticket attacks can be detected at the endpoint as follows.

    1. List all logon sessions of the system and obtain all logon IDs in hex format
    2. For each logon ID, list all Kerberos tickets that are associated with the session. Use the klist command to do so.
    3. Identify Kerberos tickets that do not match the user associated with the session. If you find any this means that they have been injected as part of a pass the ticket attack.
  - If we wanted to move the detection from the endpoint to the Domain Controller.

    The following events are of interest.

    4768 - A Kerberos authentication ticket (TGT) was requested
    4769 - A Kerberos service ticket was requested
    4770 - A Kerberos service ticket was renewed
    - 4768
4769
4770

Overpass the Hash or Pass the key

- A combination of pass the hash and pass the ticket attacks also exists called "overpass the hash" or "pass the key". Overpass the hash is also based on the Kerberos protocol.

  During an overpass the hash attack, the attacker uses an extracted NTLM hash (of another user account) to obtain a Kerberos ticket. As we have already covered this ticket can be used to access network resources.
- Detection
  - We can detect overpass the hash attempts by first looking for traces of a pass the hash attack (Event ID 4624 and Logon Type 9)* at the source host and then moving to the Domain

    Controller to identify associated 4768/4769** events (which are related to TGT and TGS requests). If we find associated 4768/4769 events, then we are dealing with an overpass the hash attack.
    - 4624
4768
4769
  - An Overpass the Hash attack can also be detected by analyzing traffic and spotting anomalies in terms of the encryption used. Attackers usually send authentication request data encrypted with RC4.

Forged Kerberos Tickets

- Golden Tickets
  - Golden Tickets are essentially forged Kerberos TGTs that can be used to request TGS tickets for any service on any computer in the domain.

    Golden Ticket creation requirements.
    - Domain Name
    - Domain SID
    - Domain KRBTGT Account NTLM password hash (Attackers should have compromised a Domain Administrator or the Domain Controller itself to get that.)
    - UserID for impersonation
  - Detection
    - Golden Tickets are quite tricky to detect. The most reliable approach to detect Golden Tickets is to look for the existence of TGS requests with no prior TGT requests before them (within a reasonable time frame). If you find any, this may be related to a golden ticket attack.

      More specific, but less reliable approaches are:

      1. Checking for the existence of anomalies in the following events. Event ID:4624 (Account Logon), Event ID: 4672 (Admin Logon) and Event ID: 4634 (Account Logoff). Specifically you may see "eo.oe.kiwl)", " · . eo.oe - ANSSI E>", a FQDN or nothing in the Account Domain field.

      2. Identifing suspicious TGT tickets by comparing the MaxTicketAge from the domain policy to the difference in the StartTime and EndTime of the cached authentication ticket.

      3. Checking for the existence of Event ID: 4769 (A Kerberos service ticket was requested) with a status code of OxIF: Integrity check on decrypted field failed. The aforementioned will occur in the case of an attacker returning after a double reset of the KRBTGT password has taken place (part of eradication of a domain compromise).

      4. Checking for tickets (like you did when defending against pass the ticket) or new logon events related to non existing users. Through a Golden Ticket attackers can impersonate anyone on the domain, including non existing users.

      5. Checking for the existence of Kerberos tickets with RC4 encryption. They aren't commonly met on modern environments. (This will happen if attackers utilized the NTLM hash while creating the Golden Ticket)
- Silver Tickets
  - A Silver Ticket is actually a valid TGS ticket. This valid TGS is forged and no communication with the DC ever occurs.

    A Silver Ticket is encrypted/signed by the service account configured with a SPN.

    The requirement for Silver Ticket creation is:

    1.A service account's password hash, if the targeted service operates under a user account [such a hash can be acquired using Kerberoast (more on that in just a bit)].
    2. A computer account's password hash, if the targeted service is hosted by a computer (such a hash can be acquired by a tool like mimikatz).
  - Detection
    - Silver Tickets are even trickier to detect than Golden Tickets.

      A reliable but heavy-going approach to detect Silver ticket is by identifying invalid Privsvr signatures within Kerberos TGS on the wire. Specifically, a Silver Ticket (forged TGS) contains a modified PAC.

      This Pac contains two signatures.
      1. Service signature (which is a checksum of the PAC encrypted with the service key)
      2. Privsvr signature (which is a checksum of the service signature encrypted with the KRBTGT key)

      Attackers performing a Silver Ticket attack most probably haven't compromised the whole domain yet and subsequently don't yet have the KRBTGT key. For this reason the Privsvr signature will most probably be invalid.

      As defenders (with access to the KRBTGT key) we can calculate the checksum of the service signature and then encrypt it with the KRBTGT key. Armed with this knowledge we can verify any Privsvr signature seen on the wire.

      The caveat here is that the abovementioned approach requires Kerberos decryption.

      An unreliable approach to detect Silver Tickets is:
      1. Check for the existence of anomalies in the following events. Event ID:4624 (Account Logon), Event ID: 4672 (Admin Logon) and Event IDl 4634 (Account Logoff). Specifically you may see "eo.oe.kiwi : "<3 eo.oe — ANSSI E>", a FQDN or nothing in the Account Domain field.

Kerberoast

- While talking about Silver Tickets, we mentioned an attacker TTP called Kerberoast, that can result in obtaining a service account's password hash.

  Kerberoast requires identifying the Service Principal Name (SPN) associated with the target service account.
- SPN Scanning
  - A service that supports Kerberos authentication must register an SPN.

    SPN scanning performs service discovery via LDAP queries to a Domain Controller. This way, no connection to every IP on the network and no port scanning are required.
  - Detection
    - SPN scanning is quite tricky to detect. Audit of LDAP events could be used, but expect a lot of noise if you decide to do so.

      1. Searching for users causing excessive 4769 events (specifically if this is done within a small time window and you notice a large number of the available domain services in the ServiceName field). At the endpoint level an AES-encrypted ticket followed by multiple RC4-encrypted tickets related to important domain services is something that should raise suspicions.

      2. Checking for the existence of Kerberos tickets with RC4 encryption. They aren't commonly met on modern environments.

      3. Creating a honey account with a Service Principal Name and then looking for 4769 events that have This accowpt in the ServiceName field.

DCSync

- Once an attacker obtains Domain or Enterprise Administrator privileges, he can act as a Domain Controller and request password data from the targeted DC.

  This attacker technique is called DCSync and enables attackers to pull password hashes (including previous ones) over the network without the interactive logon requirement and without pulling the NTDS.dit file.

  Special rights are required in order to run DCSync. Members of the 'Administrators,' 'Domain Admins' or 'Enterprise Admins' groups as well as the DC computer account itself can run DCSync.
- The interesting thing is that a normal domain user can be delegated the rights needed to extract password data. Those rights are:
  - Replicating Directory Changes
  - Replicating Directory Changes All
  - Replicating Directory Changes In
  - Filtered Set (required in some environments)
- Detection
  - DCSync performs its nefarious purposes through Active Directory replication activities. The attack specifically requests the domain controller to replicate the user credentials via GetNCChanges (Abusing MS-DRSR).

    A viable detection approach is to place all Domain Controller IP addresses in a list and configure your IDS to alert you if it detects a DsGetNCChange request from an IP that is not included in the list.

    On your right you can see two Suricata rules that detect DCSync following the approach above.

    1. The first rule will set a flowbit (drsuapi) if DRSUAPI-binding traffic is spotted on the wire
    2. The second rule will detect a DCE/RPC DsGetNCChanges request originating from an IP that is not included in the DC_SERVERS variable.
    - Suricata IDS

DCShadow

- Once an attacker obtains Domain or Enterprise Administrator privileges, he can mount a stealthy Active Directory object replication attack called DCShadow.

  DCShadow is essentially a mimikatz module that simulates the behavior of a DC, evading common security controls including SIEM solutions. The attack is similar in nature to the DCSync attack we previously covered.
- Detection
  - A DCShadow attack can be detected on the wire by spotting API like DrsAddEntry or DrsReplicaAdd being called from a machine that is not a Domain Contoller.

    DCShadow can also be detected through log analysis. Specifically, we will be able to spot objects in the Configuration partition being added or the computer object being changed.

    In addition, the following Audit Detailed Directory Service Replication events can prove useful when looking for a DCShadow attack.

    · 4928 - An Active Directory replica source naming context was established
    · 4929 - An Active Directory replica source naming context was removed

Password Spraying

- Attackers may use an identified password to launch a password spraying attack against a Domain Controller or other machines on the domain.
- Detection
  - Usually attackers perform password spraying against SMB on a domain controller. This will result in an Event ID 4625 "logon failure" being registered. The caveat here is that organizations should be able to spot such events occurring within a small time window.

    If the attackers choose to perform password spraying against LDAP no 4625 events will be logged. Event ID 4771 can help us in this case, but it requires Kerberos logging to be enabled. Look for Failure Code Oxl 8 (bad password) inside 4771 events occurring within a small time window.

    In the case of password spraying against workstations Event ID 4868 will be registered (Audit logging should be enabled to see this) Attention should also be paid to the last bad password attribute in order to discover password spraying.
    - 4625
4771

Remote User Enumeration

- There are multiple ways using which attackers perform remote (privileged) user enumeration. Some of them are:
  - Native net commands (NetSessionEnum based)
  - PowerView/BloodHound (stealthier approach)
- Detection
  - NetSessionEnum
    - First let's focus on detecting SMB Session enumeration via the NetSessionEnum method. This can easily be detected through traffic analysis.
  - PowerView and BloodHound
    - Regarding detecting PowerView and BloodHound the following approaches can be used.

      1. A big part of PowerView's (and subsequently BloodHound's) functionality comes down to LDAP queries (net.exe uses SAMR). We can detect those LDAP queries by enabling the logging of Domain Controller Event 1644. See an example on your right (upper image).

      2. The above will generate numerous events, so you may want to proceed to detecting enumeration-related LDAP queries through analyzing traffic (LDAP is a cleartext protocol). See an example of an LDAP SearchRequest on your right (image at the bottom).

      3.PowerShell's script block logging capabilities can assist in detecting PowerView's or BloodHound's PowerShell code.
      - 1644
  - honeytoken
    - A honeytoken approach can also be followed to identify remote privileged user enumeration attempts. We can audit the User and groups directory objects, just like files or folders. To enable this the "Directory Service Access" subcategory should be enabled under "DS Access."

      For remote user enumeration detection perform the following.
      1. Set up User and Group accounts (containing both regular and honeytoken user accounts that will be used for detection purposes only.)

      2. Enable the "Advanced Features" option inside the "Active Directory Users and Computers" MMC, so that the "Security" tab is now visible.

      3. Inside the "Security" tab click on "Advanced" -> "Auditing" -> "Add" and set the following properties.
      - Principal = Everyone
      - Applies to = This object only
      - Permissions = Read all properties

      From now on Event ID 4662 entries will be registered whenever one of these objects (user or group) is enumerated.
      - Directory Service Access 4662

Lateral Movement

Remote File Copy over SMB

- Attackers are known for copying files over SMB for lateral movement purposes, once they identify valid credentials.

  In the following example, we will see the attacker first connecting to the c$ share and then copying mimikatz.exe for lateral movement purposes.
- Detection
  - Remote file copy over SMB can easily be detected through traffic analysis. On your right (upper image) you can see the attacker first connecting to the c$ share and then (image at the bottom) you can see him copying mimikatz.exe starting with a "Create Request". "Get Info" is related to retrieving information about that target filesystem and "Set Info" is related to transmitting some length information and metadata at the end of the transaction. Copying the file commences from "Write Request" onwards.

    At the endpoint Event ID 5140 and Event ID 5145 can help us detect remote file copy over SMB. Windows file auditing can also enhance our visibility regarding newly "created" files.

    In addition, both Suricata and Bro have the ability to extract files transferred over SMB. They can also alert you when the C$, ADMIN$, or IPC$ shares are used.

    For further SMB network analysis refer to the following resource. https://401trg.com/an-introduction-to-smb-for-network-security-analysts/
    - 5140
5145

Remote Execution

- Remote Execution Through WMI
  - Detection
    - Remote execution through WMI can be detected by correlating Event ID 4624 with Sysmon Event ID 1.

      Specifically, you will see a Sysmon Event ID 1 related to the same Logon ID that has a ParentImage field of C:\Windows\System32\wbem\PrvSE.exe.
    - 4624
Event ID 1 - PrvSE.exe
- Remote Execution Through WinRM
  - Detection
    - Remote execution through WinRM can be detected by correlating Event ID 4624 with Sysmon Event ID 1.

      Specifically, you will see a Sysmon Event ID 1 related to the same Logon ID that has a ParentImage field of C:\Windows\System32\winrshost.exe
    - 4624
Event ID 1 - winrshost
- Remote Execution Through PS Remoting
  - Detection
    - Remote execution through PS Remoting can be detected by correlating Event ID 4624 with Sysmon Event ID 1.

      Specifically, you will see a Sysmon Event ID 1 related to the same Logon ID that has a ParentImage field of C:\Windows\System32\wsmprovhost. exe.

      In a subsequent Sysmon Event ID 1 you will notice wsmprovhost.exe starting the malicious payload.
    - 4624
Event ID 1 - wsmprovhost

Persistence

Registry Persistence

Scheduled Tasks / Cron jobs

- Windows scheduled tasks and Linux cron jobs are being abused for persistence purposes for almost a decade.

  Proactively inspecting them for traces of malicious activity is highly recommended.

WMI

- Windows Management Instrumentation (WMI) is essentially an enterprise information management framework designed to allow access to system data at scale.
- Empire WMI Persistence
  - The infamous Empire PowerShell post-exploitation framework includes a module that persists a stager (or script) using a permanent WMI subscription.

    Luckily Sysmon 6.10 added 3 new events for catching WMI Filter and Consumer Activity, as well as the binding which makes them active.
  - Detection
    - If an empire stager uses this WMI persistence module and Sysmon is deployed you will be able to see the following three (3) events.

      1. WmiEventFilter activity detected (rule: WmiEvent) Event ID 19 -
      Describes the conditions under which the payload will be triggered. Within 5 minutes of system boot in this case.

      2. WmiEventConsumer activity detected (rule: Event ID 20 - WmiEvent)
      This is where the payload resides (Base64- encoded)

      3. WmiEventComsumerToFilter (rule: WmiEvent) Event ID 21
      This is where the event consumer is bound to the event filter

      Autoruns could also have detected this through its WMI tab.

      Get-WMIObject can be used for detection and Remove-WMIObject for Eradication.
      - Event ID 19
Event ID 20
Event ID 21
    - The following event is also important when trying to track WMI activity.

      Event ID 5861 records permanent event consumer creation. The great thing about it is that it catches both the filter and the consumer.

      This Event ID can be found in "modern" systems (Win2012R2+)
      - 5861

Linux Rootkits

- Rootkits are malicious pieces of software that are able to conceal selected processes, files, network connections and directories. Rootkits are also usually equipped with stealthy backdoors to grant the attacker persistent remote access to the infected system.

  This time our analysis will be conducted through the Volatility memory forensics framework.
  - Volatility

Presented with xmind