



ZAIUX **EVO**

ZAIUX Evo - Datasheet

Version: 2.9.0 | 2024

Summary

A definition of ZAIUX Evo.....	3
Deploy, Evasion and procedures to adopt in case of EDR Detection.....	4
What to do in case both deploy methods are blocked by the EDR?.....	5
Choice of the Entry Point.....	5
Hardware and Software requirements.....	5
Automatic rollback and manual clean-up.....	5
Active Directory Health Check.....	7
Entra ID Health Check.....	8
Ransomware Attack Simulation.....	9
ZAIUX Framework integration.....	10
Attack vectors.....	11
#1 - Active Directory Enumeration.....	11
#2 - Local Privilege Escalation.....	12
#3 - Domain Privilege Escalation.....	13
#4 - Shellcode Injection.....	14
#5 - Lateral Movement.....	15
#6 - Kerberos Attacks.....	16
#7 - Hashcracking.....	16
#8 - Credential Harvesting.....	17

A definition of ZAIUX Evo

ZAIUX® Evo makes it possible, for the first time, to perform a complete and realistic simulation of an intrusion in a MS Active Directory environment with an intelligent solution, exploiting a regularly updated range of the most modern and advanced hacking techniques, run in stealth mode to emulate a human approach. Automation is managed by the DPZR™ engine that includes Machine Learning algorithms specially developed by our team of experts to emulate human intelligence, breaking down the time barrier of manual execution. Through Artificial Intelligence the adaptive algorithms, which we developed, shape the system's response according to the attack surfaces emerging from the scans, all in a fully automated way. ZAIUX® Evo is an intelligent Full Cloud platform which generates, for each assessment, an isolate sandbox, associated with an initialization package which can be directly executed from any endpoint of the target network, without installing any agent.



Software strengths



Realistic



Full Cloud



MSSP-ready



Agentless



Simple



AI-driven



Clear Reporting



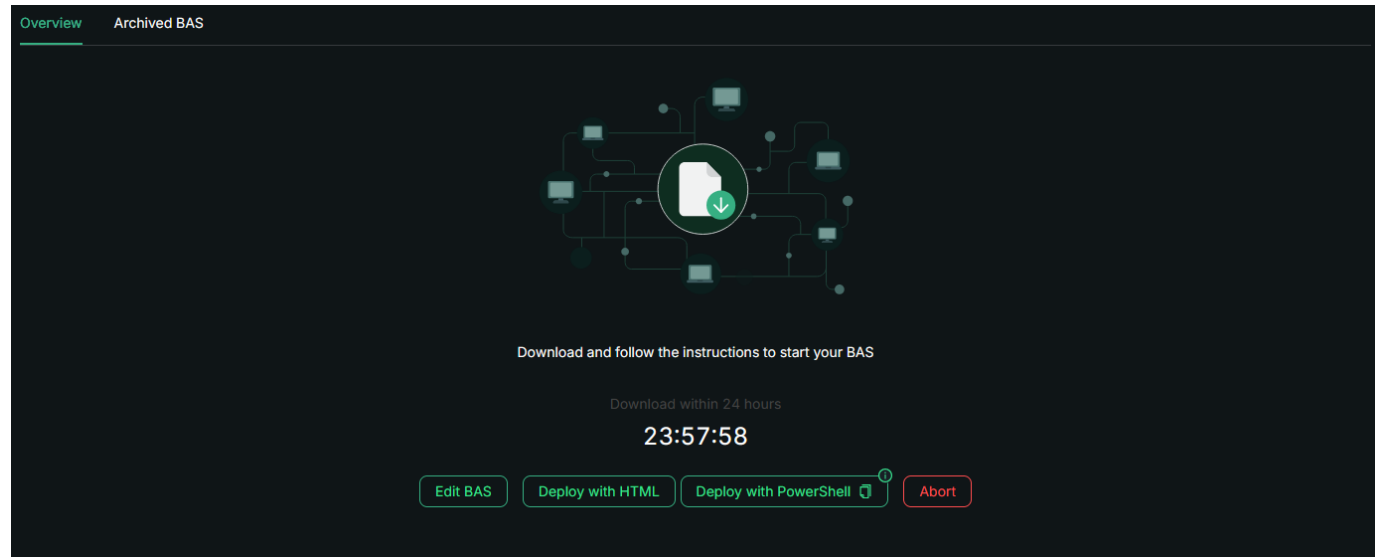
No false positives



Integrable

Deploy, Evasion and procedures to adopt in case of EDR Detection

ZAIUX Evo supports two distinct deploy methods for the Implant:



1. **HTML:** This is the most suitable deployment methodology, as in most cases it guarantees a greater stealthiness. The HTML file, as soon as it is opened in any browser on a machine that has been designated as an Entry-Point (see the next paragraph), automatically downloads an .ISO file. After mounting the ISO in a mapped drive (by double-clicking or right click → Mount) it will be sufficient to run the StartZaiux.exe executable file inside it with a simple double click

Refer to the instructions inside the HTML file for the complete procedure.

We suggest delivering the HTML file to the Entry-Point using a remote assistance system, avoiding e-mail attachments which could be marked as suspicious by an anti-phishing system. A video demonstration is available at this link: <https://www.youtube.com/watch?v=Oxahv7MjVSY>

2. **PowerShell quick command:** This methodology is offered as an alternative to the HTML file and it consists in the execution of a PowerShell command in the machines designated as Entry-Points (see the next paragraph). In the event that PowerShell is not available or the execution is blocked by the EDR, we suggest resorting to deployment method #1 (HTML).

What to do in case both deploy methods are blocked by the EDR?

The employment of ZAIUX Evo doesn't require to define any exclusion rule on its processes or executable files, as verifying the concrete efficacy of adopted defense systems is one of the main objectives of the activity. Therefore, we suggest excluding processes and/or executable files related to ZAIUX Evo exclusively in cases where the deployment is impossible or the activity is blocked right away.

We suggest monitoring the running status of ZAIUX Evo processes from the Task Manager and to enable e-mail notifications during the BAS creation phase, in order to be alerted if any problem arises during execution.

Choice of the Entry Point

There is no limit on the number of Entry-Points on which the Implant can be deployed with a single Token, as long as they all belong to the same Active Directory domain. We suggest firstly running ZAIUX Evo on Endpoints with standard Active Directory privileges, and then evaluating the execution on different Entry-Points (either PCs or servers) with different or higher user privileges.

Note: Where a forest (or multiple forest) with multiple domains is present, only the domain on which the activity is first triggered will be taken into account. It will therefore be necessary to execute as many BAS (each one at the cost of one Token) as the number of Active Directory domains the need to be targeted, possibly defining a separate Site in the Dashboard for each domain. Moreover, in case a domain is located in multiple physical infrastructures, each one having one or more Domain Controllers, we suggest to treat every infrastructure as a separate Site, executing an independent BAS on each of them.

Hardware and Software requirements

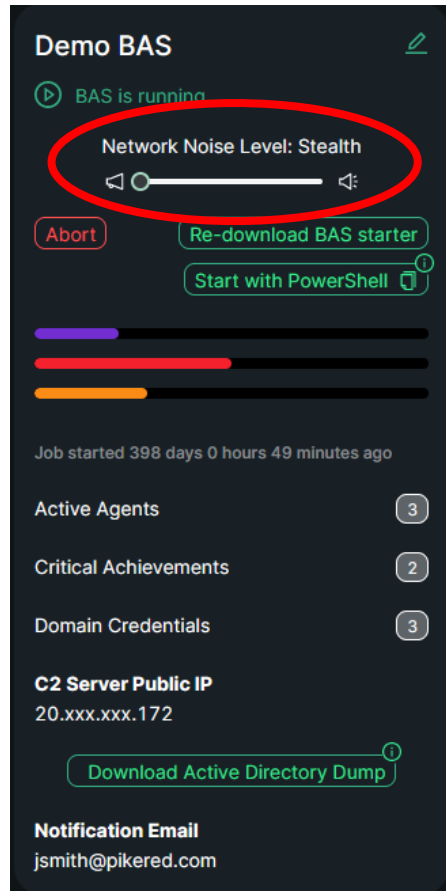
The ZAIUX Evo Implant can only be executed on Client and Server machines running a 64bit Windows OS. As of today, the following operating systems are supported: Windows Server 2016, 2019, 2022; Windows 10 and 11. There may be compatibility issues with Windows Server 2012 or Windows 8 operating systems. No compatibility is possible with earlier operating systems. There are no minimum requirements in terms of RAM, CPU and HDD.

Automatic rollback and manual clean-up

The system automatically performs the rollback of all configurations applied during the Breach & Attack Simulation activity, including the cleaning of temporary files that may be used during the test. However, in some scenarios, it may not be possible to complete the automatic rollback and manual actions by system administrators may be required. The report will underline the procedure needed to clean any traces left by ZAIUX Evo on the systems.

Note: As soon as the activity is completed, we recommend removing all files (HTML, ISO and DLL) that are used during the triggering phase on the designated Entry-Points.

Network Noise Level



After the initial AD Enumeration phase, a dashboard will be available to the user to monitor and manage the running BAS. Among the available options there is the possibility to tune the “noise level” generated by Implants inside the target network.

Typically, Network Detection & Response / IDS / IPS tools can identify the presence of a malware thanks to its peculiar behaviour known as “beaconing”, that is the tendency of the malware of sending requests to its Command & Control server on an approximately regular basis.

ZAIUX Evo, by default, starts its activity in “Stealth” mode. With this configuration the beaconing process will be masked with long-running and randomized request intervals. However, it is possible to diminish these intervals by moving the slider towards the “Normal” value at the centre or the “Noisy” value on the right.

This functionality allows to verify whether monitoring tools are tuned at suitable levels, or if configuration adjustments are needed.

Further insights at:

- <https://www.elastic.co/security-labs/identifying-beaconing-malware-using-elastic>
- <https://attack.mitre.org/tactics/TA0011/>
- <https://attack.mitre.org/techniques/T1102/002/>

Active Directory Health Check



Among the activities performed during the BAS there is a "Light Assessment" on the Hardening level of Active Directory in the target network. It is a passive procedure, and it is executed on the snapshot of the local domain taken during the initial AD Enumeration phase, which is conducted at the beginning of every Beach & Attack Simulation.

Note: Results generated by this module will not be shown on the BAS Dashboard but only in the report, which will be available to the user as soon as the BAS activities will be completed.

The Software performs the following checks:

1. Last change of the Kerberos password.
2. Users with password set to never expire.
3. Domain Admins in Protected User.
4. Presence of old Operating Systems.
5. Number of Accounts trusted to delegate to.
6. Number of computers vulnerable to unconstrained delegation.
7. Last usage of the native Administrator account.
8. Schema Admins group.
9. Admins without the flag "This account is sensitive and cannot be delegated".
10. Minimum Password Length Policy for Domain Accounts.
11. Maximum Password Age Policy for Domain Accounts.

Entra ID Health Check



Microsoft Entra ID

Among the activities performed during the BAS there is a “Light Assessment” on the Hardening level of the Azure Entra ID Tenant. To achieve this, ZAIUX Evo executes a series of atomic checks within the Entra ID tenant and suggests recommendations to enhance the security posture.

NOTA: I risultati di questo modulo non saranno mostrati sulla Dashboard del BAS ma esclusivamente sulla reportistica, che sarà a disposizione del cliente non appena l'attività sarà stata completata

1. **Users with Global Administrator role:** <https://learn.microsoft.com/en-us/entra/identity/role-based-access-control/permissions-reference#global-administrator>
2. **High number of high privileged roles:** <https://learn.microsoft.com/en-us/entra/identity/role-based-access-control/best-practices#6-limit-the-number-of-privileged-role-assignments-to-less-than-10>
3. **Permission to consent application from non-verified publishers:** <https://learn.microsoft.com/en-us/entra/identity/enterprise-apps/configure-user-consent?pivots=portal>
4. **Conditional Access Policies usage:** <https://learn.microsoft.com/en-us/entra/identity/conditional-access/overview>
5. **Dynamic Groups and Guest invite usage:** <https://learn.microsoft.com/en-us/entra/identity/users/groups-dynamic-membership>

Ransomware Attack Simulation

Create BAS

Tag (OPTIONAL)

Notification Email (OPTIONAL)

Receive notifications for this job at jsmith@pikered.com

Ransomware Attack Simulator (OPTIONAL)

Simulate a ransomware attack in this BAS

ZAIUX Framework Team - Server Fallback (OPTIONAL)

IP/Hostname Port

Check Team-Server Reachability

Cancel Create BAS

The Token also includes a Ransomware simulation function, which can be deactivated by disabling the associated checkbox during the configuration of the Breach & Attack Simulation (BAS).

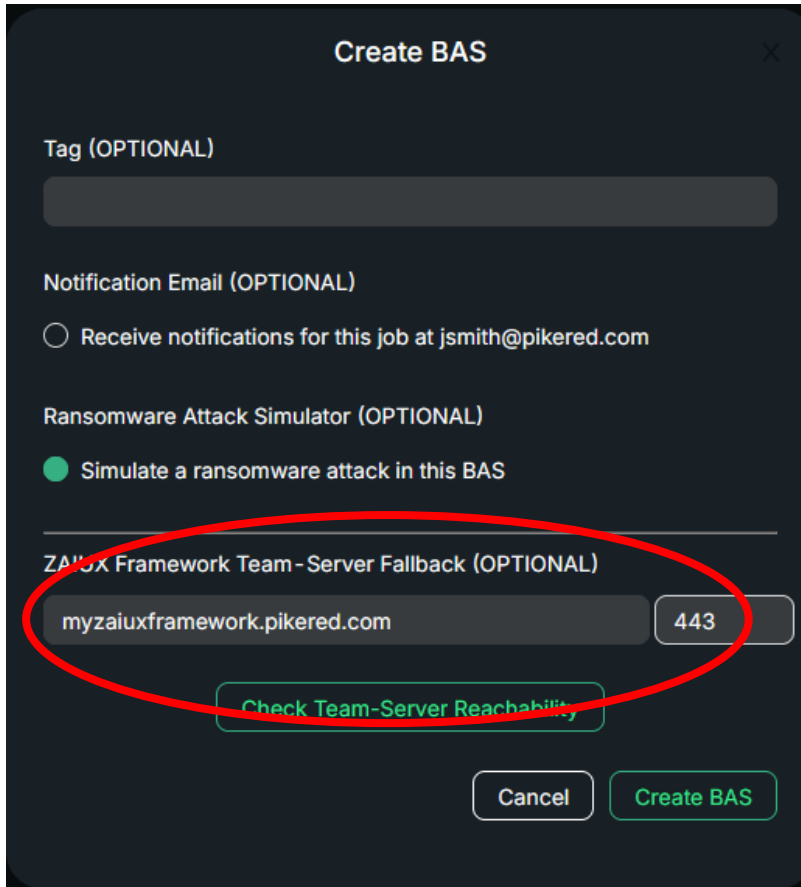
The Ransomware simulation activity is performed in a completely safe way, not impacting the content of the file servers, as ad-hoc files will be uploaded to avoid interacting with the real files.

The files that will be employed in the Ransomware simulation include some of the most common extensions, such as PNG, DOCX, PDF, etc... Moreover, a TXT file will be uploaded containing the typical instructions for ransom payment.

The activity will be performed on all shared folders (SMB) and the local C:\ of the compromised computers and servers, excepting the Domain Controllers. All the encrypted files will be left in the folders as a counterproof of the successful encryption. The affected paths will be underlined in the report, so that system administrators can proceed with manual clean-up.

The results of this module will not be shown in the BAS Dashboard, instead they will be exclusively included in the report, which will be available to the user as soon as the activity ends.

ZAIUX Framework integration



Create BAS

Tag (OPTIONAL)

Notification Email (OPTIONAL)

Receive notifications for this job at jsmith@pikered.com

Ransomware Attack Simulator (OPTIONAL)

Simulate a ransomware attack in this BAS

ZAIUX Framework Team - Server Fallback (OPTIONAL)

myzaiuxframework.pikered.com 443

Check Team-Server Reachability

Cancel Create BAS

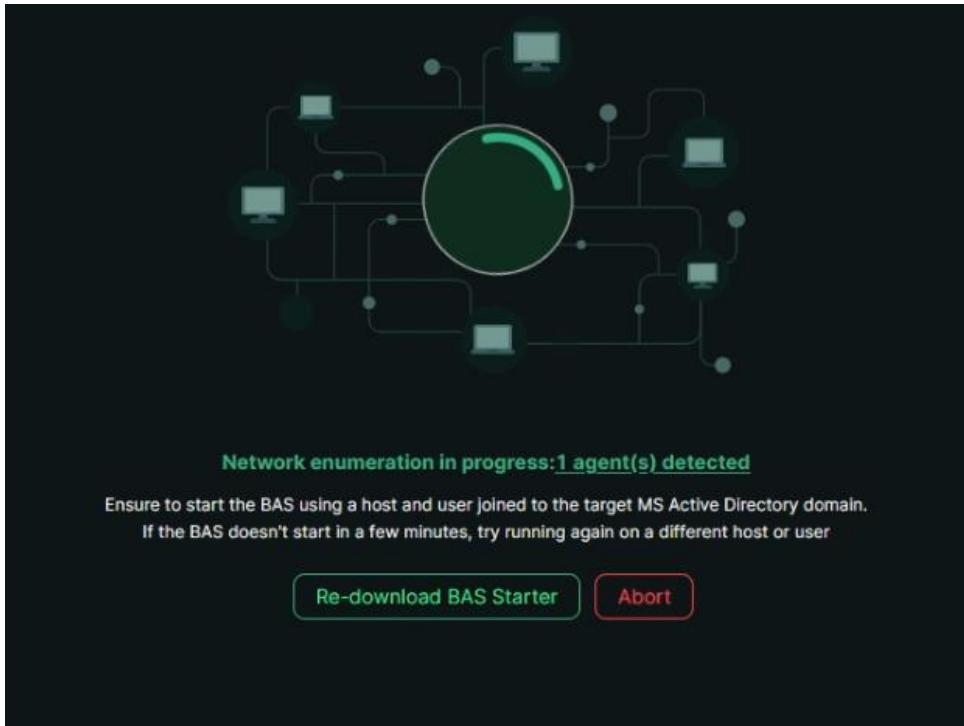
If the customer holds a valid [ZAIUX Framework](#) license, during the configuration phase of the BAS it is possible to specify the parameter of their Team Server (URL/IP and Port) as well as the license key, in order to connect the two Software solutions.

This integration offers the possibility to the Red Team to manually continue the activity autonomously performed by ZAIUX Evo. This is made possible by the automatic migration of all Implants generated by ZAIUX Evo, inheriting their command history, including the results, in an on-premises or Cloud instance of ZAIUX Framework.

A demonstration video, showing the procedure step-by-step, is available at the following link: <https://www.youtube.com/watch?v=Cr1VbMeiokA>

Attack vectors

#1 - Active Directory Enumeration



The first phase of every Breach & Attack Simulation (BAS) performed by ZAIUX Evo consists in the enumeration of the Active Directory domain to which the user that has triggered the execution belongs. The attack has been designed to be as realistic as possible and it employs the tools typically used by Threat Actors.

Specifically, an open-source tool commonly employed for this kind of operations is used: <https://github.com/BloodHoundAD/SharpHound>

Considering the programming language used to write this tool (C#), some preliminary steps are needed before uploading the software in the memory of the ZAIUX Evo process.

These operations include patching two different defense and telemetry systems: [AMSI](#) and [ETW](#) (User-Land only).

The bypass of these two monitoring interfaces can be performed both through a direct patching of DLL memory (amsi.dll and ntdll.dll) and through Hardware Breakpoints, without touching library memory. This behaviour is typically employed by Threat Actors as well, which need to avoid telemetry from being collected by the EDR installed in the machine. The enumeration includes the collection of the following Active Directory objects: computers, users, groups and members, Group Policy Objects (GPOs), Access Control Entries (ACEs).

The Active Directory Dump is then exfiltrated towards the ZAIUX Evo Cloud platform, in the temporary, demilitarized sandbox specifically created for the running BAS. At this point the dump is removed from the machine hard drive and the Cloud proceeds with the import and analysis, to identify possible attack paths that an attacker may exploit to raise their privileges and move laterally in the domain.

#2 – Local Privilege Escalation

Following an enumeration of current privileges, registry keys and configured policies, ZAIUX Evo may detect various types of attack vectors which could let a potential attacker raise their privileges inside the local machine.

A Local Privilege Escalation attack, if performed successfully, will let ZAIUX Evo run an Implant with HIGH INTEGRITY privileges on the machine and hence execute further attacks requiring higher privileges.

The techniques implemented to achieve this goal are:

1. **UAC Bypass #1 – Fodhelper:** If the user is a member of Local Administrators it is possible to raise the privileges and gain High Integrity. References: <https://www.elastic.co/security-labs/exploring-windows-uac-bypasses-techniques-and-detection-strategies>
2. **UAC Bypass #2 – SSPI Datagram Context:** If the user is a member of Local Administrators it is possible to raise the privileges and gain High Integrity. References: <https://splintercod3.blogspot.com/p/bypassing-uac-with-sspi-datagram.html>
3. **AlwaysInstallElevated Policy Abuse:** If this attribute is enabled in the system registry, every user in the machine can install MSI packages with administrative privileges. An attacker could abuse this functionality to raise their privileges, References: <https://learn.microsoft.com/en-us/windows/win32/msi/alwaysinstallelevated>
4. **LAPS Misconfiguration Abuse:** If LAPS (<https://learn.microsoft.com/it-it/archive/blogs/secguide/remote-use-of-local-accounts-laps-changes-everything>) was not suitably configured, it could be exploited by a potential attacker to raise their privileges or move laterally inside the network.
5. **Resource-Based Constrained Delegation:** Exploiting the WebClient service it is possible to perform an NTLM relay and thus execute a Local Privilege Escalation allowing to jump from a standard Active Directory user to NT AUTHORITY\SYSTEM. References: <https://www.ired.team/offensive-security-experiments/active-directory-kerberos-abuse/resource-based-constrained-delegation-ad-computer-object-take-over-and-priviledged-code-execution>
6. **GPP Credential Extraction:** In some legacy Active Directory scenarios created in the past, it is possible to extract plaintext credentials from the Group Policy Preferences and leverage those credentials to escalate privileges locally.
7. **Unquoted Service Path:** This vulnerability occurs when a service is created with a path that contains spaces and it's not properly enclosed between double quotes. ZAIUX Evo is able to detect this misconfiguration and elevate its privileges.

Note: the above-mentioned attack vectors are composed of two phases: a Discovery phase and an Exploitation phase. The first phase verifies the possibility to perform the attack, while the second phase is the actual attack execution. In case both phases are successfully executed, two distinct rows will be shown in the report and in the Dashboard.

#3 – Domain Privilege Escalation

ZAIUX Evo includes various attack vectors to perform a Privilege Escalation across the Active Directory domain. Some attacks include a Discovery and an Exploitation phase and will thus appear as distinct rows in the report and in the Dashboard.

1. **Spooler Service Enumeration:** It verifies whether the print spooler is enabled on Domain Controllers. That service may be exploited in various attack scenarios. References: <https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/printers-spooler-service-abuse>
2. **Machine Account Quota Enumeration:** The ms-DS-MachineAccountQuota attribute indicates how many computer accounts can be configured by each user. It is advisable to force this value to 0 to prevent various attack scenarios. References: <https://www.netspi.com/blog/technical/network-penetration-testing/machineaccountquota-is-useful-sometimes/>
3. **Domain Password Spraying:** The accounts identified as critical and/or useful for the simulation activity undergo a password spraying attack with threshold values sufficient to avoid the risk of blocking users. References: <https://attack.mitre.org/techniques/T1110/003/>
4. **Active Directory Certificate Services Enumeration:** It verifies the presence of vulnerable certificates in ADCS. References: https://specterops.io/wp-content/uploads/sites/3/2022/06/Certified_Pre-Owned.pdf
5. **Certified Pre-Owned ESC1:** It exploits a misconfiguration of certificate templates in ADCS to obtain higher privileges in the domain. Riferimenti: https://specterops.io/wp-content/uploads/sites/3/2022/06/Certified_Pre-Owned.pdf
6. **Certified Pre-Owned ESC3:** It exploits a misconfiguration of certificate templates in ADCS to obtain higher privileges in the domain. Riferimenti: https://specterops.io/wp-content/uploads/sites/3/2022/06/Certified_Pre-Owned.pdf
7. **Certified Pre-Owned ESC4:** It exploits a misconfiguration of certificate templates in ADCS to obtain higher privileges in the domain. Riferimenti: https://specterops.io/wp-content/uploads/sites/3/2022/06/Certified_Pre-Owned.pdf
8. **ClearText Password Property:** In some cases, it is possible to get the passwords of user accounts from Active Directory attributes of those objects. This module checks for the presence of passwords and verifies their validity before employing them. References: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-adts/f3adda9f-89e1-4340-a3f2-1f0a6249f1f8
9. **SQL User Impersonation:** ZAIUX Evo is capable of identifying if it's possible to impersonate high privileged users, exploiting Microsoft SQL Servers instances discovered during the Enumeration phase.

#4 – Shellcode Injection

Typically, an attacker resorts to shellcode injection when there the necessity to create multiple instances of the malware in the same machine. This necessity can rise if a backup/support Implant is needed or when one wants to create new processes and/or threads to monitor the environment and gather useful information for the subsequent phases.

ZAIUX Evo employs two distinct Shellcode Injection techniques to reach different goals:

1. **“Classic” Shellcode Injection** (Indirect Syscalls): NtOpenProcess → NtAllocateVirtualMemory → NtProtectVirtualMemory → NtCreateThreadEx

Demonstration of the classical shellcode injection which exploits the Native APIs provided by the operating system itself. To validate the response from the perspective of the EDR, an Indirect variant can be adopted, in order to have a clean Return Address in the Stack Trace.

ZAIUX Evo employs this technique when it needs to create an additional Implant on the same machine to perform concurrent, interacting attacks, or to create a backup Implant to be used in case the main Implant is blocked by defense systems.

2. **Spawn-and-Inject** (Indirect Syscalls): NtCreateUserProcess → NtAllocateVirtualMemory → NtWriteVirtualMemory → NtProtectVirtualMemory → NtQueueApcThread → NtResumeThread

Demonstration of how it is possible to create a new process with the main thread in a suspended state, to hijack it toward a subsequently allocated shellcode. In order to validate the response from the perspective of the EDR, ZAIUX Evo employs an Indirect variant (see above), PPID Spoofing and the PROCESS_CREATION_MITIGATION_POLICY_BLOCK_NON_MICROSOFT_BINARIES_ALWAYS_ON attribute, which prevents some EDRs from injecting their monitoring library in the process.

ZAIUX Evo employs this technique to inject modules capable of monitoring the execution of certain processes by the user, in order to extract sensible information such as hashes or plaintext credentials. Additional information can be found in the “Credential/Token Stealing, Dump & Extraction” section.

#5 – Lateral Movement

ZAIUX Evo includes different functionalities which allow its payload to expand in the network:

1. **WinRM + AppDomain Injection:** ZAIUX EVO can employ mechanisms typically exploited by attackers to execute malicious code through binaries marked as safe. This technique is known as [AppDomain](#) Injection and it employs a specifically created dll, a .config file and a binary .NET legitimate file which are uploaded on the remote machine through the SMB protocol. The legitimate executable file is then run via the WinRM protocol.
2. **Upload via SMB + WinRM:** In case the computer targeted by the Lateral Movement is not configured to communicate with the Internet, ZAIUX Evo will autonomously upload a DLL which will then be executed through rundll32.exe, launching the command via WinRM. The library consists in a modified ZAIUX Evo payload, capable of communicating via SMB with the originary Implant which in turn communicates with the Team-Server via HTTPS. This behaviour is known as Pivoting, and it allows to compromise those systems which wouldn't be normally able to communicate with endpoints outside the network.
3. **Remote Desktop Protocol:** If valid credentials are identified to establish an RDP connection, these can be employed to execute a command on the remote machine exploiting the RDP protocol. This attack vector is thus efficiently masked by a behaviour which would be considered as normal in most cases.
4. **GPO Abuse:** ZAIUX Evo can identify GPOs vulnerable to potentially unwanted changes, so as to deploy a Payload on multiple machines (up to 10 at a time).
5. **Abuse SQL for Lateral Movement:** ZAIUX Evo is capable of leverage existing SQL Server Instances to move laterally. The attack path leverages the loading of a special DLL written in C# by loading it into a Stored-Procedure, resulting in a file-less attack.

#6 – Kerberos Attacks

1. **Kerberoasting:** If, from Active Directory analysis, some Service Principal Names are identified relating to services such as IIS, SQL or EXCHANGE, it is possible to employ this technique, which allows to any user in the domain to request a Ticket for those specific SPNs. Cracking will then be attempted on the obtained hash to try and recover the plaintext password.
2. **AS-Rep Roasting:** If, from Active Directory analysis, user accounts having the “Do not require kerberos preauthentication” flag enabled, it is possible to directly request their hash. Cracking will then be attempted on the obtained hash to try and recover the plaintext password.
3. **Unconstrained Delegation:** ZAIUX Evo can identify which machines are subject to Unconstrained Delegation. It is possible to force every computer (including the Domain Controller) to authenticate to these machines, in order to extract the Kerberos tickets and impersonate users and services in the domain.
4. **S4u2Self Abuse:** The S4U2Self extension was introduced to support Kerberos Constrained Delegation, and it allows a service to obtain a service ticket for itself on a user’s behalf. However, some parts of the ticket are not encrypted, so it is possible to modify them. By changing the valute of the Service Principal Name, ZAIUX Evo can impersonate various user categories in the compromised machines.
5. **Overpass-the-Hash / Pass-the-Ticket** In alternative to the most common and abused Pass-The-Hash, it is possible to resort to less invasive techniques to inject valid credentials or tickets in memory, aiming at impersonating different users or authenticating on critical services.
6. **Make token:** If, during the BAS activity, a plaintext password is detected for a domain user, it is possible to impersonate that user in order to perform subsequent Lateral Movement actions.

#7 – Hashcracking

In order to avoid consuming hardware resources (RAM and CPU) of the compromised machines to perform hashcracking, every hash is exfiltrated to our Cloud, where its scalability is employed to try and recover the plaintext password. The adopted hashcracking methodology combines different techniques based on common password dictionaries, permutations, key-walking and dynamically generated strings based on the context. If the hash is successfully cracked, it will be shown (masked) in the report. The password will then be employed by ZAIUX Evo for subsequent operations, such as Lateral Movement.

#8 – Credential Harvesting

ZAIUX Evo employs various modules able to extract credentials from process memory or configuration files of different programs which may be present in the compromised computers.

1. **Browser Credentials Dump:** A common user practice consists in saving credentials in browsers. This module is able to extract the saved credentials from the most common browsers, namely Chrome, Edge, Firefox and Internet Explorer. These credentials are then analyzed to find valid passwords for the Active Directory domain.
2. **Backup Credential Dump:** Backup tools employed in Enterprise environments are critical targets for potential attackers aiming at compromising company backups. ZAIUX Evo can run attacks aimed at extracting the credentials employed in managed backup processes, discovering how deep an attacker could go after compromising a backup machine.
3. **LSASS Memory Dump:** The aim of this module is to perform a dump of the lsass.exe process, exfiltrate it and parse it, looking for plaintext credentials or valid hashes. The lsass.exe process is in fact responsible of the management of passwords, access tokens and policies. Given its sensitive nature, it is often targeted by attackers aiming at quickly raising their privileges.
The attack vector is composed of 3 distinct parts:
 - System process enumeration: Firstly, all processes in the operating system are identified, and the lsass.exe PID (Process ID) is detected.
 - Possible protection of the lsass.exe process through PPL is verified: <https://www.elastic.co/blog/protecting-windows-protected-processes>
If PPL is not active, it will be shown in the report as a medium-severity vulnerability.
 - If PPL is not active, the process memory dump can be performed, and it is encrypted before exfiltrating it.
4. **Steal Token:** An Implant with HIGH INTEGRITY privileges can enumerate every process in the system, to identify access tokens assigned to them. After identifying the target process, it is possible to duplicate and impersonate its access token, to interact with local or remote resource and, potentially, execute Lateral Movement operations.
5. **RDP Credential Stealing:** ZAIUX Evo is able to monitor the Remote Desktop processes being opened by the user, in order to steal the plaintext credentials as they are entered during the login phase.
6. **Sysadmins' tool Abuse:** Typically, sysadmins (and other users as well) install tools on their computers to be able to interact with services exposed by other servers. These services can include SSH, SFTP, FTP, RDP etc. To save time, credentials are typically saved, sometimes without setting a master password to

protect them. A common practice of Threat-Actors is to identify the easiest way to gather credentials in plain text. The purpose of this module is therefore to identify credentials that are not properly protected.

7. **Credential Vault** This module leverages operating system-specific APIs to verify the ability to extract plaintext credentials that are stored in the Windows Vault. These credentials could be login passwords for SMB shared folders, mailbox access credentials, or other types of services.
8. **Kerberos Ticket Enumeration:** If possible, ZAIUX Evo will attempt an extraction of cached Kerberos tickets on the computer. These tickets can then be used by attacks such as Pass-The-Ticket or S4u2self.
9. **Credential Phishing:** Thanks to the interaction with the operating system's APIs, it is possible to assess the resilience of users and targeted phishing attacks that require the input of one's own or others' credentials.
10. **Linked SQL Server Credential Discovery:** ZAIUX Evo is able to determine if it is possible to extract credentials from Linked Servers in the SQL instances identified during the Enumeration phase.
11. **Sensitive Data Discovery:** ZAIUX Evo, by leveraging a series of Windows APIs such as FindFirstFileA, FindNextFileA, and CreateFileA, is capable of autonomously identifying files that may contain sensitive information within the entire File System. These files may include memory dumps, command-line history, or configuration files like Web.config. If an appropriate technique exists to exploit these credentials, it will be used to move laterally or escalate privileges.

#9 – Entra ID Techniques

ZAIUX Evo can leverage the following techniques to enumerate users, groups, and roles within the customer's Entra ID tenant. This information is then also used by the system to complete the Entra ID Health Check described in the relevant section.

- 1. Steal Application Access Tokens:** Access tokens can be found in the memory of processes that use them to access data on behalf of the user, such as OUTLOOK.EXE or MS-TEAMS.EXE. Windows APIs like `VirtualQueryEx`, `NtQueryVirtualMemory`, `ReadProcessMemory`, and `NtReadProcessMemory` can be abused to read the access tokens from the memory of remote processes. The tokens are then exploited by ZAIUX Evo to access specific resources within the Entra ID tenant.
- 2. E-mail Collection:** Zaiux EVO uses a previously obtained access token to attempt to access a user's e-mails. The requested data includes only the subject and sender of the latest e-mails in the inbox folder.
- 3. Steal Primary Refresh Token Cookie:** A Primary Refresh Token (PRT) is present on a device when an Entra ID user signs into an Entra-joined or hybrid-joined device. For devices registered to Entra ID, the token is present if there is a secondary work account. The PRT allows the user to use SSO (Single Sign-On). ZAIUX Evo can extract the user's Primary Refresh Token from the memory of the `lsass.exe` process and can then use it to authenticate to Entra ID, bypassing MFA requirements.