

mavinject.exe İncelemesi

Sistemde çalıştırılan komut;

“mavinject.exe ((Get-Process lsass).Id) /INJECTRUNNING C:\Windows\System32\vbscript.dll”

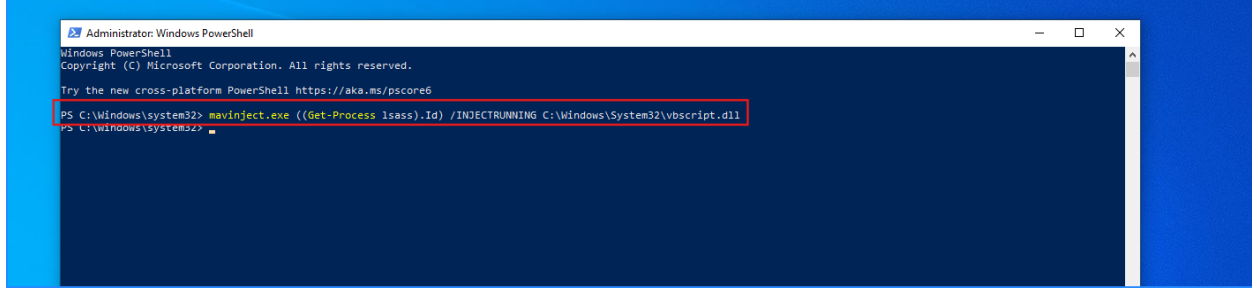


Figure 1 - Komut Powershell ile Çalıştırıldı

Çalıştırıldıktan sonra incelenen Windows Powershell logları;

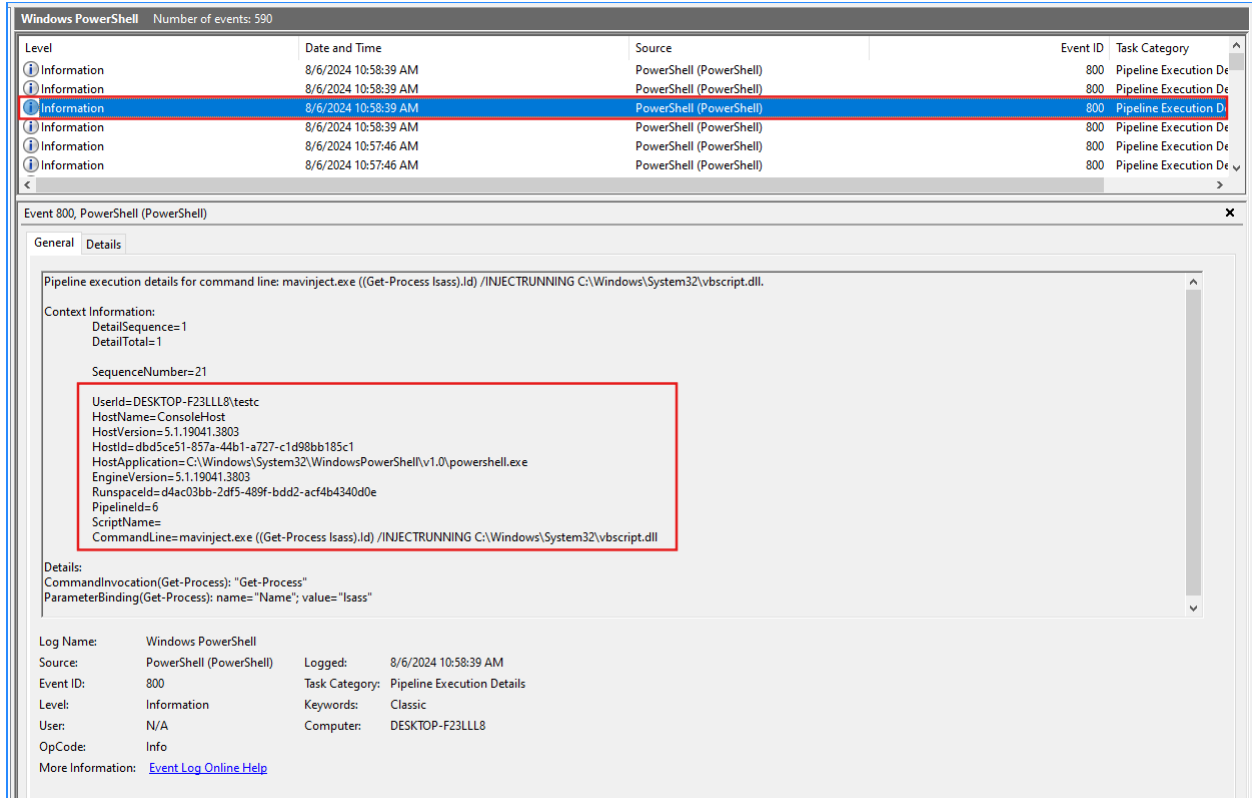


Figure 2 - PowerShell Logları

Çalıştırıldıktan sonra incelenen Sysmon logları

Level	Date and Time	Source	Event ID	Task Category
Information	8/6/2024 11:00:37 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	8/6/2024 10:59:45 AM	Sysmon	22	Dns query (rule: DnsQuery)
Information	8/6/2024 10:59:04 AM	Sysmon	3	Network connection detected (rule: ...)
Information	8/6/2024 10:58:39 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	8/6/2024 10:58:21 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	8/6/2024 10:58:09 AM	Sysmon	8	CreateRemoteThread detected (rule: ...)
Information	8/6/2024 10:58:09 AM	Sysmon	1	Process Create (rule: ProcessCreate)

Event 1, Sysmon

General Details

Process Create:
RuleName: -
UtcTime: 2024-08-06 17:58:39.622
ProcessGuid: {6835382f-644f-66b2-0802-000000000700}
ProcessId: 2240
Image: C:\Windows\System32\mavinject.exe
FileVersion: 10.0.19041.3636 (WinBuild.160101.0800)
Description: Microsoft Application Virtualization Injector
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: mavinject64.exe
CommandLine: \"C:\Windows\system32\mavinject.exe\" 656 /INJECTRUNNING C:\Windows\System32\vbscript.dll
CurrentDirectory: C:\Windows\system32\
User: DESKTOP-F23LLL8\testc
LogonGuid: {6835382f-5d11-66b2-a8b0-010000000000}
LogonId: 0x1B0A8
TerminalSessionId: 1
IntegrityLevel: High
Hashes: MD5=D9A4F566A8AD07702A9B212EFE922D25, SHA256=46A9C5234B3CC5352B5CC562B240AED8304CF993A4625C69174CF746CB1FA8D, IMPHASH=429058796B83BC005DB1F177F75548C
ParentProcessGuid: {6835382f-6418-66b2-f501-000000000700}
ParentProcessId: 6720
ParentImage: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
ParentCommandLine: \"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe\"

Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon
Event ID: 1
Level: Information
User: SYSTEM
OpCode: Info
More Information: [Event Log Online Help](#)

Logged: 8/6/2024 10:58:39 AM
Task Category: Process Create (rule: ProcessCreate)
Keywords:
Computer: DESKTOP-F23LLL8

Figure 3 - Sysmon Logları

“mavinject.exe ((Get-Process lsass).Id) /INJECTRUNNING

C:\Windows\System32\vbscript.dll” genel anlamda ilk olarak incelendiğinde mavinject.exe isimli bir executable dosya aracılığı ile LSASS (Local Security Authority Subsystem Service) servisinin yani lsass.exe’nin altına “vbscript.dll” kod parçacığını eklediği yorumlanabilir.

LSASS, Windows'ta kimlik doğrulama ve güvenlik politikaları işlemlerini yürütür.

vbscript.dll, Visual Basic Script dilini çalıştırmak için kullanılan bir .dll yani kod parçacıdır.

Process Hacker aracı ile lsass.exe incelendiğinde Modules kısmında altında çalışan dll arasında vbscript.dll'in de bulunduğu görüntüleniyor.

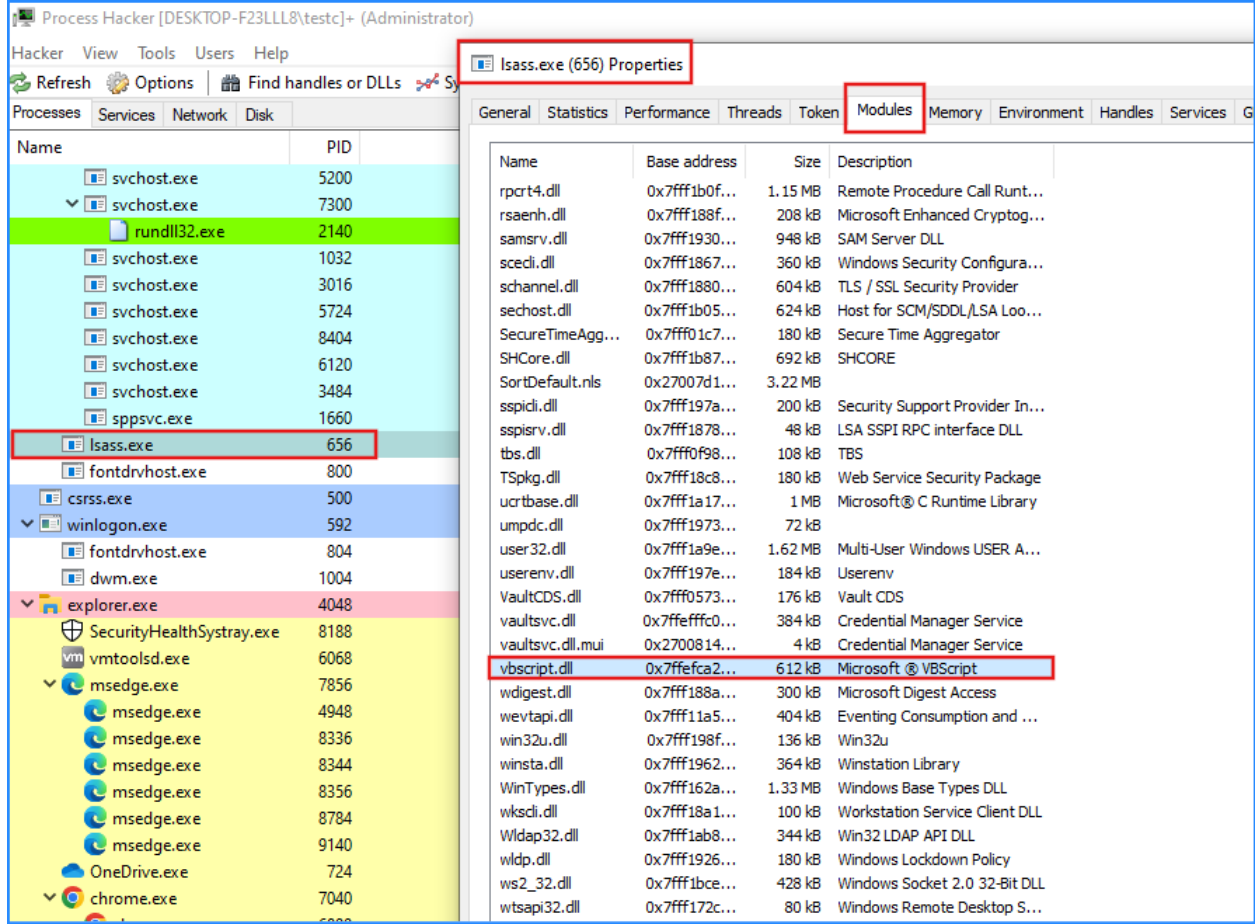
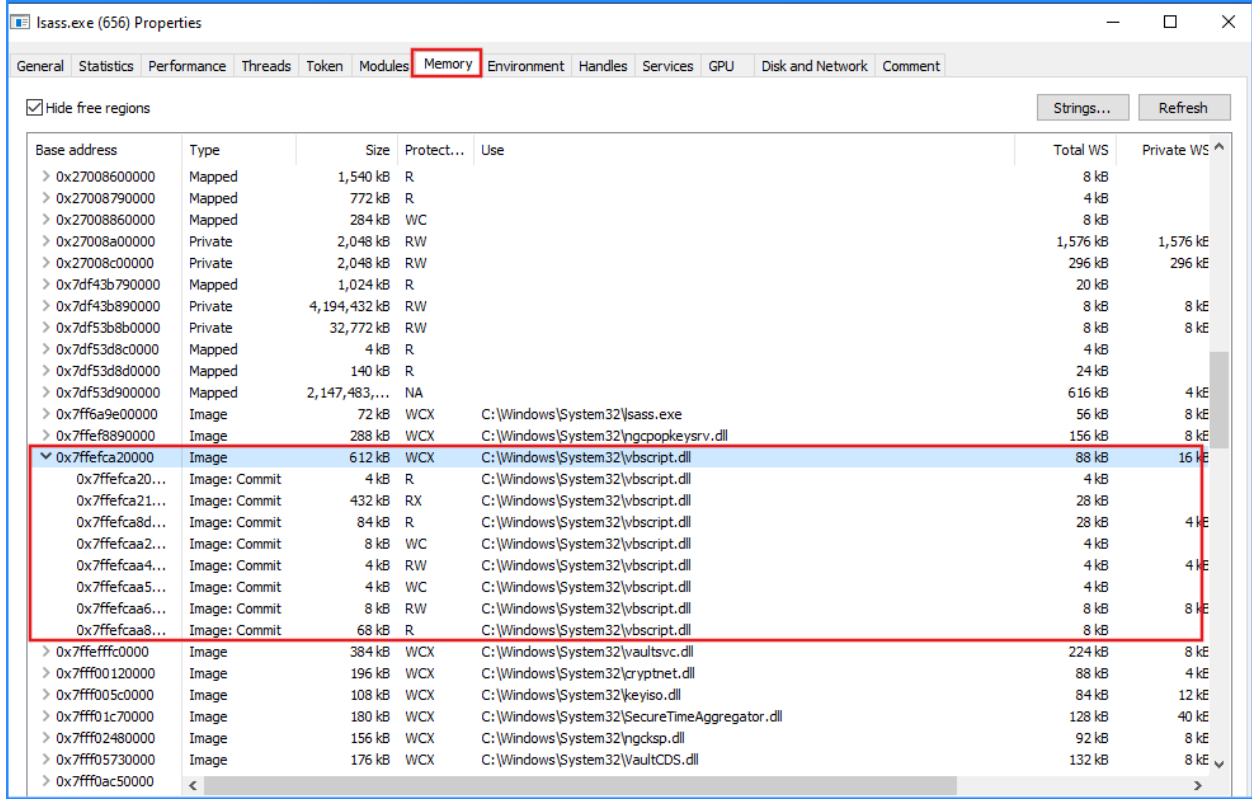


Figure 4 - vbscript.dll, lsass.exe Altında Çalışıyor

Bu aşamada saldırganlar vbscript.dll komut dosyasının içerisine zararlı kod parçacıkları ekleyerek lsass.exe'den kimlik doğrulama bilgilerinin çalınması gibi durumlar ile faydalanabilir ve kritik sonuçlar doğurabilir.

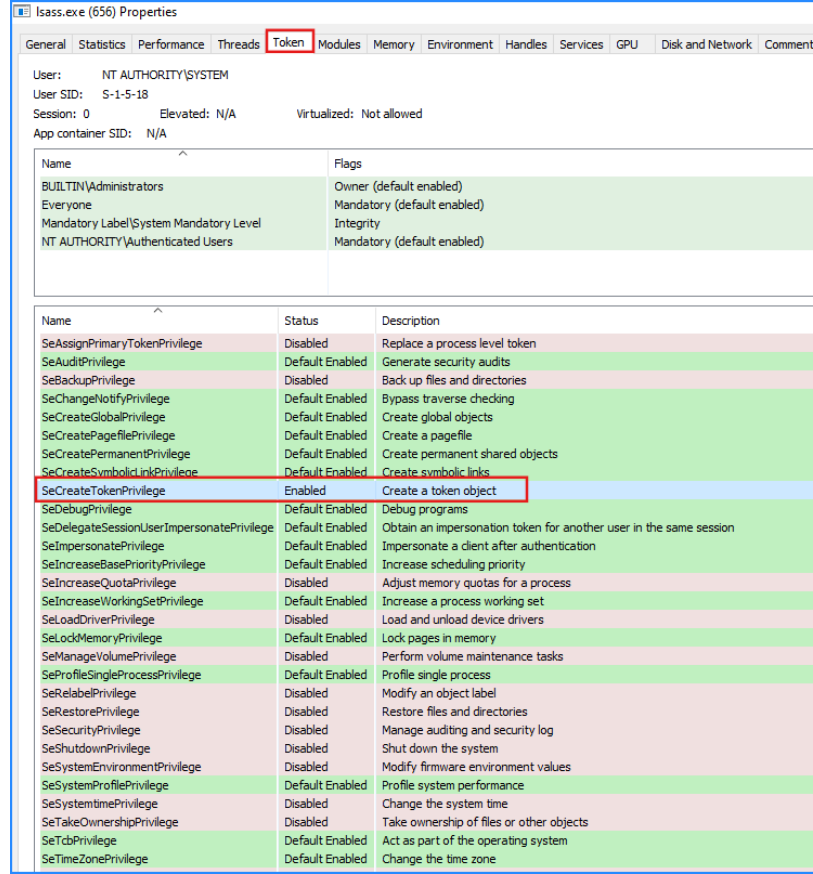
Memory sekmesi incelendiğinde “0x7ffefca20000” numarası ile vbscript.dll bulunduğu ve WCX yani Write, Copy, Executable haklarına sahip olduğu görünmektedir. Bununla birlikte 8 adet alt process oluşturmuş ve bunlara çeşitli haklar tanınmış olduğu görüntüleniyor.



Base address	Type	Size	Protect...	Use	Total WS	Private WS
> 0x27008600000	Mapped	1,540 kB	R		8 kB	
> 0x27008790000	Mapped	772 kB	R		4 kB	
> 0x27008860000	Mapped	284 kB	WC		8 kB	
> 0x27008a00000	Private	2,048 kB	RW		1,576 kB	1,576 kB
> 0x27008c00000	Private	2,048 kB	RW		296 kB	296 kB
> 0x7df43b790000	Mapped	1,024 kB	R		20 kB	
> 0x7df43b890000	Private	4,194,432 kB	RW		8 kB	8 kB
> 0x7df53b8b0000	Private	32,772 kB	RW		8 kB	8 kB
> 0x7df53d8c0000	Mapped	4 kB	R		4 kB	
> 0x7df53d8d0000	Mapped	140 kB	R		24 kB	
> 0x7df53d900000	Mapped	2,147,483,...	NA		616 kB	4 kB
> 0x7ff6a9e00000	Image	72 kB	WCX	C:\Windows\System32\lsass.exe	56 kB	8 kB
> 0x7ffef8890000	Image	288 kB	WCX	C:\Windows\System32\ngcpopkeysrv.dll	156 kB	8 kB
0x7ffefca20000	Image	612 kB	WCX	C:\Windows\System32\vbscript.dll	88 kB	16 kB
0x7ffefca21...	Image: Commit	4 kB	R	C:\Windows\System32\vbscript.dll	4 kB	
0x7ffefca21...	Image: Commit	432 kB	RX	C:\Windows\System32\vbscript.dll	28 kB	
0x7ffefca8d...	Image: Commit	84 kB	R	C:\Windows\System32\vbscript.dll	28 kB	4 kB
0x7ffefcaa2...	Image: Commit	8 kB	WC	C:\Windows\System32\vbscript.dll	4 kB	
0x7ffefcaa4...	Image: Commit	4 kB	RW	C:\Windows\System32\vbscript.dll	4 kB	4 kB
0x7ffefcaa5...	Image: Commit	4 kB	WC	C:\Windows\System32\vbscript.dll	4 kB	
0x7ffefcaa6...	Image: Commit	8 kB	RW	C:\Windows\System32\vbscript.dll	8 kB	8 kB
0x7ffefcaa8...	Image: Commit	68 kB	R	C:\Windows\System32\vbscript.dll	8 kB	
> 0x7ffefcc00000	Image	384 kB	WCX	C:\Windows\System32\vaultsvc.dll	224 kB	8 kB
> 0x7ff001200000	Image	196 kB	WCX	C:\Windows\System32\cryptnet.dll	88 kB	4 kB
> 0x7ff005c00000	Image	108 kB	WCX	C:\Windows\System32\keyiso.dll	84 kB	12 kB
> 0x7ff01c700000	Image	180 kB	WCX	C:\Windows\System32\SecureTimeAggregator.dll	128 kB	40 kB
> 0x7ff024800000	Image	156 kB	WCX	C:\Windows\System32\ngcksp.dll	92 kB	8 kB
> 0x7ff057300000	Image	176 kB	WCX	C:\Windows\System32\VaultCDS.dll	132 kB	8 kB
> 0x7ff0ac500000						

Figure 5 - vbscript.dll Memory Alanı

Token sekmesi incelendiğinde “SeCreateTokenPrivilege” yetkisinin diğerleri “Default Enabled” olarak gelmesine rağmen sadece “Enabled” statüsünde olduğu görüntülendi, bu durumda default olarak gelmemiş ve sonradan etkinleştirilebilmiş olabileceğinden şüphelenilmiştir.



Isass.exe (656) Properties		
General Statistics Performance Threads Token Modules Memory Environment Handles Services GPU Disk and Network Comment		
User: NT AUTHORITY\SYSTEM		
User SID: S-1-5-18		
Session: 0 Elevated: N/A Virtualized: Not allowed		
App container SID: N/A		
Name	Flags	
BUILTIN\Administrators	Owner (default enabled)	
Everyone	Mandatory (default enabled)	
Mandatory Label\System Mandatory Level	Integrity	
NT AUTHORITY\Authenticated Users	Mandatory (default enabled)	
Name	Status	Description
SeAssignPrimaryTokenPrivilege	Disabled	Replace a process level token
SeAuditPrivilege	Default Enabled	Generate security audits
SeBackupPrivilege	Disabled	Back up files and directories
SeChangeNotifyPrivilege	Default Enabled	Bypass traverse checking
SeCreateGlobalPrivilege	Default Enabled	Create global objects
SeCreatePagefilePrivilege	Default Enabled	Create a pagefile
SeCreatePermanentPrivilege	Default Enabled	Create permanent shared objects
SeCreateSymbolicLinkPrivilege	Default Enabled	Create symbolic links
SeCreateTokenPrivilege	Enabled	Create a token object
SeDebugPrivilege	Default Enabled	Debug programs
SeDelegateSessionUserImpersonatePrivilege	Default Enabled	Obtain an impersonation token for another user in the same session
SeImpersonatePrivilege	Default Enabled	Impersonate a client after authentication
SeIncreaseBasePriorityPrivilege	Default Enabled	Increase scheduling priority
SeIncreaseQuotaPrivilege	Disabled	Adjust memory quotas for a process
SeIncreaseWorkingSetPrivilege	Default Enabled	Increase a process working set
SeLoadDriverPrivilege	Disabled	Load and unload device drivers
SeLockMemoryPrivilege	Default Enabled	Lock pages in memory
SeManageVolumePrivilege	Disabled	Perform volume maintenance tasks
SeProfileSingleProcessPrivilege	Default Enabled	Profile single process
SeRelabelPrivilege	Disabled	Modify an object label
SeRestorePrivilege	Disabled	Restore files and directories
SeSecurityPrivilege	Disabled	Manage auditing and security log
SeShutdownPrivilege	Disabled	Shut down the system
SeSystemEnvironmentPrivilege	Disabled	Modify firmware environment values
SeSystemProfilePrivilege	Default Enabled	Profile system performance
SeSystemtimePrivilege	Disabled	Change the system time
SeTakeOwnershipPrivilege	Disabled	Take ownership of files or other objects
SeTcbPrivilege	Default Enabled	Act as part of the operating system
SeTimeZonePrivilege	Default Enabled	Change the time zone

Figure 6 – SeCreateTokenPrivilege

SeCreateTokenPrivilege, “bir işlemin yeni bir güvenlik tokenı oluşturmaya izin veren bir Windows güvenlik ayrıcalığıdır. Bu ayrıcalık, özellikle işlem tokenlarını oluşturma yeteneği sağlar ve dolayısıyla, bir işlemin yeni güvenlik kimlik bilgileri oluşturmaya izin verir.” Olarak tanımlanıyor. Yani vbscript.dll kod parçacığı daha sonradan bu izini Enabled konumuna getirip zararlı etkinliklerde bulunma sürecini ilerletmiş olabilir.

MITRE ATT&CK

Bu senaryoya genel olarak bakıldığında bir Process Injection saldırısı olduğu anlaşılabilir. Windows sistemlerinde çalışan lsass.exe processine vbscript.dll'inin Process Injection tekniği ile enjekte edildiği saptanmıştır. MITRE ATT&CK Framework'e bakıldığında "T1055 – Process Injection" aşamasında değerlendirilebilir.

Process Injection

Sub-techniques (12) ▼

Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process.

There are many different ways to inject code into a process, many of which abuse legitimate functionalities. These implementations exist for every major OS but are typically platform specific.

More sophisticated samples may perform multiple process injections to segment modules and further evade detection, utilizing named pipes or other inter-process communication (IPC) mechanisms as a communication channel.

ID: T1055

Sub-techniques: [T1055.001](#), [T1055.002](#), [T1055.003](#), [T1055.004](#), [T1055.005](#), [T1055.008](#), [T1055.009](#), [T1055.011](#), [T1055.012](#), [T1055.013](#), [T1055.014](#), [T1055.015](#)

ⓘ **Tactics:** [Defense Evasion](#), [Privilege Escalation](#)

ⓘ **Platforms:** Linux, Windows, macOS

ⓘ **Defense Bypassed:** Anti-virus, Application control

Contributors: Anastasios Pingios; Christiaan Beek, @ChristiaanBeek; Ryan Becwar

Version: 1.3

Created: 31 May 2017

Figure 7 - MITRE & T1055

Process Injection alt tekniklerine göz atacak olursak senaryonun bir DLL Injection olduğu varsayılarak T1055.001 kategorisinde değerlendirilebileceği saptanmıştır.

Process Injection: Dynamic-link Library Injection

Other sub-techniques of Process Injection (12) ▼

Adversaries may inject dynamic-link libraries (DLLs) into processes in order to evade process-based defenses as well as possibly elevate privileges. DLL injection is a method of executing arbitrary code in the address space of a separate live process.

DLL injection is commonly performed by writing the path to a DLL in the virtual address space of the target process before loading the DLL by invoking a new thread. The write can be performed with native Windows API calls such as `VirtualAllocEx` and `WriteProcessMemory`, then invoked with `CreateRemoteThread` (which calls the `LoadLibrary` API responsible for loading the DLL).^[1]

Variations of this method such as reflective DLL injection (writing a self-mapping DLL into a process) and memory module (map DLL when writing into process) overcome the address relocation issue as well as the additional APIs to invoke execution (since these methods load and execute the files in memory by manually performing the function of `LoadLibrary`).^{[2][1]}

Another variation of this method, often referred to as Module Stomping/Overloading or DLL Hollowing, may be leveraged to conceal injected code within a process. This method involves loading a legitimate DLL into a remote process then manually overwriting the module's `AddressOfEntryPoint` before starting a new thread in the target process.^[3] This variation allows attackers to hide malicious injected code by potentially backing its execution with a legitimate DLL file on disk.^[4]

Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via DLL injection may also evade detection from security products since the execution is masked under a legitimate process.

ID: T1055.001

Sub-technique of: [T1055](#)

ⓘ **Tactics:** [Defense Evasion](#), [Privilege Escalation](#)

ⓘ **Platforms:** Windows

ⓘ **Permissions Required:** User

ⓘ **Defense Bypassed:** Anti-virus, Application control

Contributors: Boominathan Sundaram

Version: 1.3

Created: 14 January 2020

Last Modified: 11 August 2023

[Version Permalink](#)

Figure 8 - T1055.001 - DLL Injection

SIGMA RULE

title: Detecting mavinject.exe Process/DLL Injection into lsass.exe

status: experimental

description: mavinject.exe'nin gerçekleştirdiği vbscript.dll Process/DLL Injection Tespiti

references:

- Florian Roth from Nextron Systems

author: Doğukan KAYA

date: 2024/08/06

tags:

- attack.process_injection
- attack.t1055
- attack.dynamic-link_library_injection
- attack.t1055.001

logsource:

product: windows

service: Sysmon

category: process_creation

detection:

selection:

Image|endswith: 'mavinject.exe'

CommandLine|contains:

- '/INJECTRUNNING '
- 'C:\Windows\System32\vbscript.dll
- ' lsass.exe'

condition: selection

fields:

- Image
- CommandLine
- ProcessId
- ParentProcessId

falsepositives:

level: high

Process/DLL Injection Saldırısı

Bu saldırı göz önünde bulundurulduğunda, saldırgan vbscript.dll dosyasını, mavinject.exe isimli executable dosyasını çalıştırarak lsass.exe processinin içerisine enjekte etmeyi hedeflemiştir. Bu saldırıda zararlı .dll dosyası hedef sistemde lsass.exe altında çalışarak lsass.exe'nin tuttuğu ve oluşturduğu tüm processleri, kritik bilgileri ele geçirebilir. Bu DLL injection sayesinde hedef sistemde shell alabilir ve administrator yetkileri ile hedef windows sistemde istediği komutları çalıştırabilir. Böylelikle zararlı .dll başarılı bir şekilde enjekte edildikten sonra saldırganın bir sonraki adımı LSASS'in tuttuğu kullanıcı doğrulama bilgilerinin elde edilmesi, hedef sisteme Administrator yetkileri ile sahip olup, içeriden kritik verileri alıp ya da şifreleyip bir Ransomware saldırısı girişiminde bulunabilir. Hedef sisteme backdoor yerleştirip, fark edilmeden uzun bir süre kritik verileri içeriden çıkarabilir. Bu gibi saldırılarda kurumlar ya da kişiler Windows cihazlarda çalışan ve şüphelenilmeyen bir process ile (lsass.exe) kritik verilerini kaybedebilir, Fidyeye yazılımı saldırılarına kurban olabilir ve ciddi sonuçlar doğurabilir. Kullanılan SIEM ürünlerine gerekli kuralların detaylı bir şekilde yazılması ve test edilmesi gerekmektedir.

Bu aşama MITRE ATT&CK Framework'ünde T1055 ve Alt Teknik olan T1055.001 ile ele alınır. Bu teknikleri kullanan bir çok hacker/APT grubu bulunmaktadır.

Bu saldırılar arasında **ZEUS PANDA** örnek olarak verilebilir. MITRE'ye göre;

Zeus Panda, banka bilgilerini ve diğer hassas kimlik bilgilerini çalmak için tasarlanmış bir Truva atıdır. Zeus Panda'nın orijinal kaynak kodu 2011'de sızdırıldı ve bu, tehdit aktörlerinin kaynak kodunu yeni zararlı yazılım varyantları oluşturmak için kullanmasına olanak sağladı. Genellikle Windows XP'den Windows 10'a kadar olan Windows işletim sistemlerini hedef alır.

Kullandığı teknikler arasında T1055 – Process Injection'da bulunur. Sistemdeki işlemlerin kontrolünü sağladıktan sonra gerekli koşullar sağlanıyorsa işleme zararlı kodu enjekte eder.

<https://securityintelligence.com/panda-is-one-hungry-bear-a-heavyweight-banking-trojan-rolls-into-brazil/>