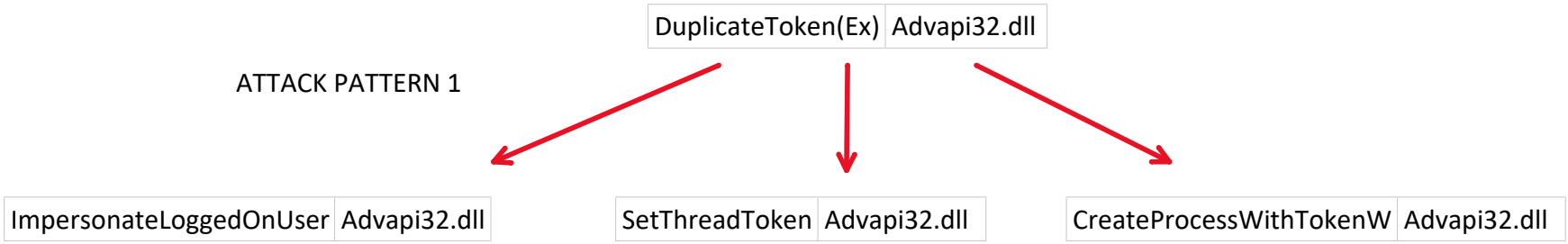
Access token manupulation

Wednesday, January 15, 2020 12:55 PM

Product	Detection Tracker	Subject Process	Subject commandline	Subject Integrity level	Parent Process	Parent Commandline	Process action	Parent Integrity Level	Registry
ATP	Telemetry showed svchost.exe executed with the seclogon command-line argument and a subsequent elevated powershell.exe process, indicating token manipulation (tainted by parent alert on a suspicious PowerShell command-line generated for the svchost.exe invocation of powershell.exe with an encoded script).	powershell	encoded command	High	svchost	svchost.exe -k netsvcs -p -s seclogon	Load Image	system	
ATP	Telemetry showed svchost.exe as a high integrity process from SYSTEM and subsequent cmd.exe process running as user George (tainted by the parent alert on suspicious process injection into lsass.exe). Svchost.exe was executed with seclogon command-line argument indicating token manipulation.	cmd.exe	reg query command	High	svchost	svchost.exe -k netsvcs -p -s seclogon	Load Image	system	

Access Token Manipulation

Windows uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it belongs to someone other than the user that started the process. When this occurs, the process also takes on the security context associated with the new token.



On attack pattern 1 attacker having access to some service of low privilege can use duplicate token function in windows of a privilege user session with the process id then he has option to

1-Impersonate the user of hight privilege - **ImpersonateLoggedOnUser function in Advapi32.dll**

2-Apply the privileged token impersonated to the attackers process - **SetThreadToken function in** Advapi32.dll

3-Create a process with a high privilege token which is duplicated - **CreateProcessWithTokenW function in** Advapi32.dll

LogonUser	Advapi32.dll

ATTACK PATTERN 2

SetThreadToken	Advapi32.dll

On attack Pattern 2 if the attacker have a user name and password do not see a user online with privilege he may not have a access token to duplicate
1- He can use LogonUser make the user login to the system
2 - Then set the thread token created into the attackers process using SetThreadToken

Refer

https://attack.mitre.org/techniques/T1134/

1 - Monitoring TTP :

1 - Process tree from system level integrity to High level Integrity
2 - Execution of below mentioned API inside non windows files
DuplicateToken(Ex)
ImpersonateLoggedOnUser
SetThreadToken
CreateProcessWithTokenW
LogonUser
SetThreadToken

2 - VT Hunt

Api = [
DuplicateToken,

3 - Write Yara RULE

Convert these windows functions into hex
 write rule for matching file having atleast 3/5 patterns

ImpersonateLoggedOnUser,
SetThreadToken ,
CreateProcessWithTokenW,
LogonUser,
]

For values in API:
String = "imports:" + "values"

FUNCTIONS

```
BOOL SetThreadToken(
  PHANDLE Thread,
  HANDLE  Token
);
```

```
BOOL DuplicateTokenEx(
  HANDLE                hExistingToken,
  DWORD                 dwDesiredAccess,
  LPSECURITY_ATTRIBUTES       lpTokenAttributes,
  SECURITY_IMPERSONATION_LEVEL ImpersonationLevel,
  TOKEN_TYPE              TokenType,
  PHANDLE               phNewToken
);
```

```
BOOL ImpersonateLoggedOnUser(
  HANDLE hToken
);
```

```
BOOL LogonUserA(
  LPCSTR  lpszUsername,
  LPCSTR  lpszDomain,
  LPCSTR  lpszPassword,
  DWORD   dwLogonType,
  DWORD   dwLogonProvider,
  PHANDLE phToken
);
```

```
BOOL CreateProcessWithTokenW(
  HANDLE              hToken,
  DWORD               dwLogonFlags,
  LPCWSTR             lpApplicationName,
  LPWSTR              lpCommandLine,
  DWORD               dwCreationFlags,
  LPVOID              lpEnvironment,
  LPCWSTR             lpCurrentDirectory,
  LPSTARTUPINFOW      lpStartupInfo,
  LPPROCESS_INFORMATION lpProcessInformation
);
```