Map-View Code Injection

by creating Views on Sections of Memory and Mapping them to remote processes

Basic Concepts

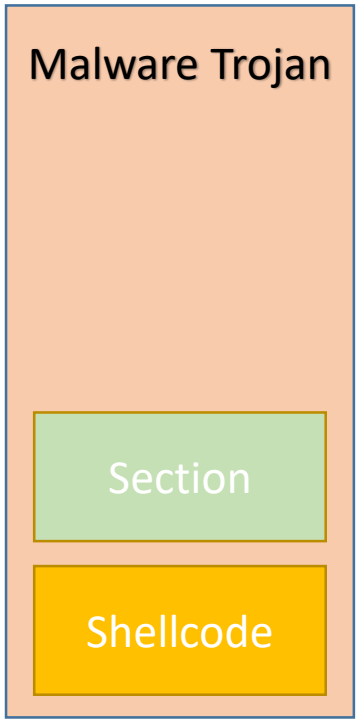
- Inter Process Communication (IPC) via Mapping-View techniques
- By sharing memory between 2 processes
- The Malware shares its memory with a Target Process
- Then, the Malware executes the shared memory remotely via the Target Process

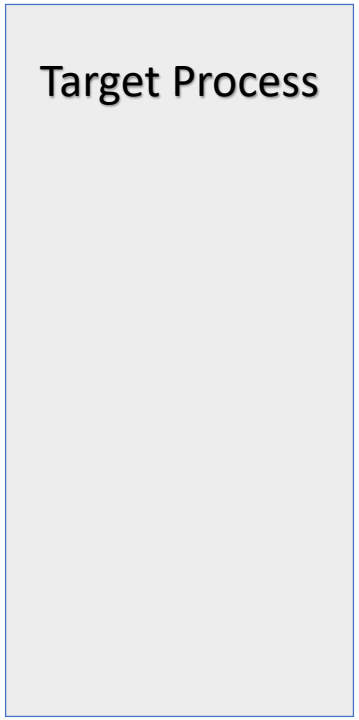
Mechanism of Map-View Code Injection

Mechanism of Map-View Code Injection

1. Create Section

NtCreateSection





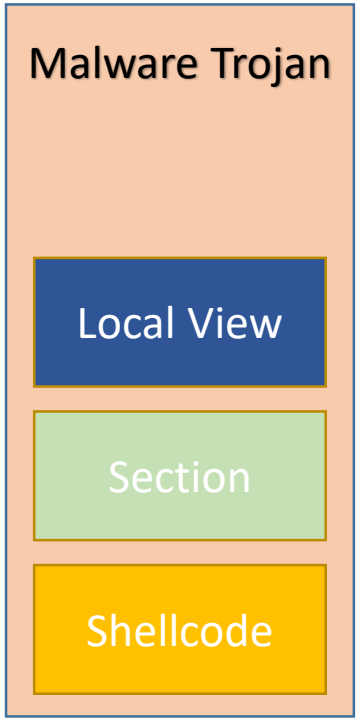
Mechanism of Map-View Code Injection

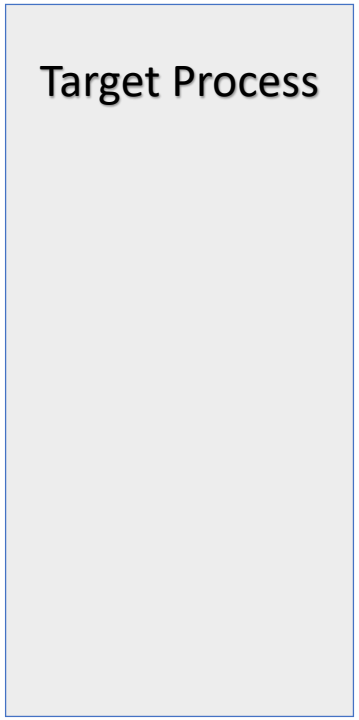
1. Create Section

NtCreateSection

2. Create Local View

NtMapViewOfSection





Mechanism of Map-View Code Injection

1. Create Section

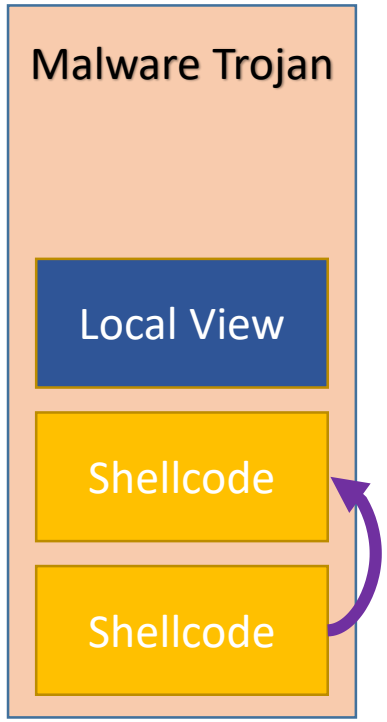
NtCreateSection

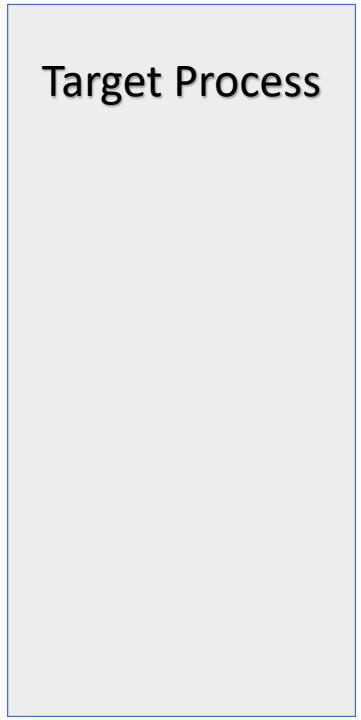
2. Create Local View

NtMapViewOfSection

3. Copy Shellcode to Section

memcpy





Mechanism of Map-View Code Injection

1. Create Section

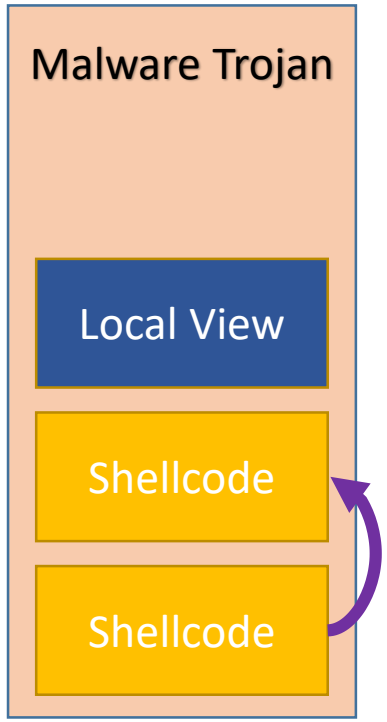
NtCreateSection

2. Create Local View

NtMapViewOfSection

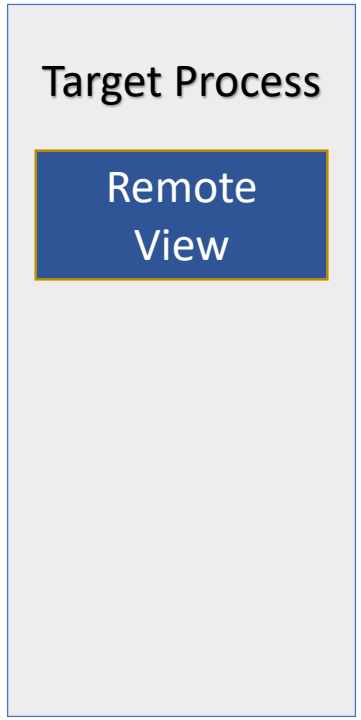
3. Copy Shellcode to Section

memcpy

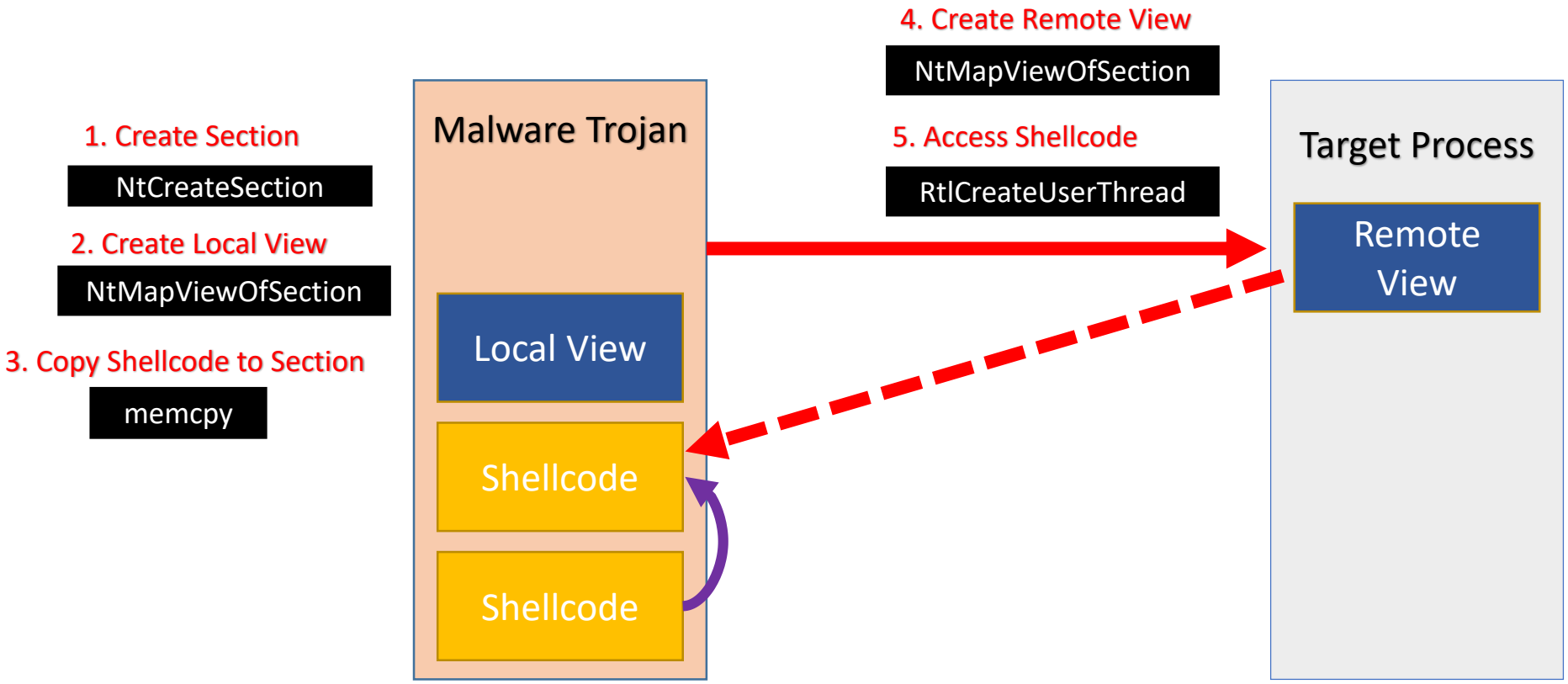


4. Create Remote View

NtMapViewOfSection



Mechanism of Map-View Code Injection



Advantages & Disadvantages

Of Map-View Code Injection

Advantages

- No need to use
  - VirtualAllocEx
  - WriteProcessMemory
- The above calls are classic tell-tale signs of process injection which AV can detect
- By sharing memory, we make it appear like a legitimate remote process is executing the shellcode
- The Target Process acts as a proxy for the Malware
- The Malware runs the shellcode via the Target Process
- More Stealthy

Disadvantages

- It makes use of the API NtMapViewOfSection which may be monitored by AV

References

https://hakin9.org/mapping-injection-just-another-windows-process-injection/

Thank you