# TTP Detail – T1055.015

## TTP Information

Name: ListPlanting

Description: Adversaries may abuse list-view controls to inject malicious code into hijacked processes in order to evade process-based defenses as well as possibly elevate privileges. ListPlanting is a method of executing arbitrary code in the address space of a separate live process.(Citation: Hexacorn Listplanting) Code executed via ListPlanting may also evade detection from security products since the execution is masked under a legitimate process.  
  
List-view controls are user interface windows used to display collections of items.(Citation: Microsoft List View Controls) Information about an application's list-view settings are stored within the process' memory in a <code>SysListView32</code> control.  
  
ListPlanting (a form of message-passing "shatter attack") may be performed by copying code into the virtual address space of a process that uses a list-view control then using that code as a custom callback for sorting the listed items.(Citation: Modexp Windows Process Injection) Adversaries must first copy code into the target process’ memory space, which can be performed various ways including by directly obtaining a handle to the <code>SysListView32</code> child of the victim process window (via Windows API calls such as <code>FindWindow</code> and/or <code>EnumWindows</code>) or other [Process Injection](https://attack.mitre.org/techniques/T1055) methods.  
  
Some variations of ListPlanting may allocate memory in the target process but then use window messages to copy the payload, to avoid the use of the highly monitored <code>WriteProcessMemory</code> function. For example, an adversary can use the <code>PostMessage</code> and/or <code>SendMessage</code> API functions to send <code>LVM\_SETITEMPOSITION</code> and <code>LVM\_GETITEMPOSITION</code> messages, effectively copying a payload 2 bytes at a time to the allocated memory.(Citation: ESET InvisiMole June 2020)   
  
Finally, the payload is triggered by sending the <code>LVM\_SORTITEMS</code> message to the <code>SysListView32</code> child of the process window, with the payload within the newly allocated buffer passed and executed as the <code>ListView\_SortItems</code> callback.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Kill Chain Phases

**•** mitre-attack: defense-evasion

**•** mitre-attack: privilege-escalation

## Malware

* InvisiMole