Shared Modules

Adversaries may execute malicious payloads via loading shared modules. Shared modules are executable files that are loaded into processes to provide access to reusable code, such as specific custom functions or invoking OS API functions (i.e., Native API).

Adversaries may use this functionality as a way to execute arbitrary payloads on a victim system. For example, adversaries can modularize functionality of their malware into shared objects that perform various functions such as managing C2 network communications or execution of specific actions on objective.

The Linux & macOS module loader can load and execute shared objects from arbitrary local paths. This functionality resides in `dlfcn.h` in functions such as `dlopen` and `dlsym`. Although macOS can execute `.so` files, common practice uses `.dylib` files.[1][2][3][4]

The Windows module loader can be instructed to load DLLs from arbitrary local paths and arbitrary Universal Naming Convention (UNC) network paths. This functionality resides in `NTDLL.dll` and is part of the Windows Native API which is called from functions like `LoadLibrary` at run time.[5]

ID: T1129

Sub-techniques: No sub-techniques

ⓘ

Tactic: Execution

ⓘ

Platforms: Linux, Windows, macOS

Contributors: Stefan Kanthak

Version: 2.2

Created: 31 May 2017

Last Modified: 12 October 2023

Version Permalink

Procedure Examples

ID	Name	Description
S0373	Astaroth	Astaroth uses the LoadLibraryExW() function to load additional modules. [6]
S0438	Attor	Attor's dispatcher can execute additional plugins by loading the respective DLLs.[7]
S0520	BLINDINGCAN	BLINDINGCAN has loaded and executed DLLs in memory during runtime on a victim machine.[8]
S0415	BOOSTWRITE	BOOSTWRITE has used the DWriteCreateFactory() function to load additional modules.[9]
S1039	Bumblebee	Bumblebee can use `LoadLibrary` to attempt to execute GdiPlus.dll.[10]
S0673	DarkWatchman	DarkWatchman can load DLLs.[11]
S0567	Dtrack	Dtrack contains a function that calls `LoadLibrary` and `GetProcAddress`.[12]
S0377	Ebury	Ebury is executed through hooking the keyutils.so file used by legitimate versions of `OpenSSH` and `libcurl`.[13]
S0661	FoggyWeb	FoggyWeb's loader can call the `load()` function to load the FoggyWeb dll into an Application Domain on a compromised AD FS server.[14]
S0032	gh0st RAT	gh0st RAT can load DLLs into memory.[15]
S0203	Hydraq	Hydraq creates a backdoor through which remote attackers can load and call DLL functions.[16][17]
S0607	KillDisk	KillDisk loads and executes functions from a DLL.[18]
S0455	Metamorfo	Metamorfo had used AutoIt to load and execute the DLL payload.[19]
S0352	OSX_OCEANLOTUS.D	For network communications, OSX_OCEANLOTUS.D loads a dynamic library (`.dylib` file) using `dlopen()` and obtains a function pointer to execute within that shared library using `dlsym()`.[4]
S0501	PipeMon	PipeMon has used call to `LoadLibrary` to load its installer. PipeMon loads its modules using reflective loading or custom shellcode.[20]
S0196	PUNCHBUGGY	PUNCHBUGGY can load a DLL using the LoadLibrary API.[21]
S1078	RotaJakiro	RotaJakiro uses dynamically linked shared libraries (`.so` files) to execute additional functionality using `dlopen()` and `dlsym()`.[3]
S0603	Stuxnet	Stuxnet calls LoadLibrary then executes exports from a DLL.[22]
S0467	TajMahal	TajMahal has the ability to inject the `LoadLibrary` call template DLL into running processes.[23]
S1154	VersaMem	VersaMem relied on the Java Instrumentation API and Javassist to dynamically modify Java code existing in memory.[24]

Mitigations

ID	Mitigation	Description
M1038	Execution Prevention	Identify and block potentially malicious software executed through this technique by using application control tools capable of preventing unknown modules from being loaded.

Detection

ID	Data Source	Data Component	Detects
DS0011	Module	Module Load	Monitor shared module loading, focusing on .dll, .so, and .dylib files, and look for suspicious paths or abnormal module loads that deviate from system norms. Limiting module loads to trusted directories, such as `%SystemRoot%` and `%ProgramFiles%` on Windows, may protect against module loads from unsafe paths.
DS0009	Process	OS API Execution	Monitor API calls such as LoadLibrary (Windows) or dlopen (Linux/macOS) that load shared modules.

References

1. Apple. (2012, July 23). Overview of Dynamic Libraries. Retrieved September 7, 2023.
2. Wheeler, D. (2003, April 11). Shared Libraries. Retrieved September 7, 2023.
3. Alex Turing, Hui Wang. (2021, April 28). RotaJakiro: A long live secret backdoor with 0 VT detection. Retrieved June 14, 2023.
4. Erye Hernandez and Danny Tsechansky. (2017, June 22). The New and Improved macOS Backdoor from OceanLotus. Retrieved September 8, 2023.
5. Microsoft. (2023, April 28). What is a DLL. Retrieved September 7, 2023.
6. Salem, E. (2019, February 13). ASTAROTH MALWARE USES LEGITIMATE OS AND ANTIVIRUS PROCESSES TO STEAL PASSWORDS AND PERSONAL DATA. Retrieved April 17, 2019.
7. Hromcova, Z. (2019, October). AT COMMANDS, TOR-BASED COMMUNICATIONS: MEET ATTOR, A FANTASY CREATURE AND ALSO A SPY PLATFORM. Retrieved May 6, 2020.
8. US-CERT. (2020, August 19). MAR-10295134-1.v1 – North Korean Remote Access Trojan: BLINDINGCAN. Retrieved August 19, 2020.
9. Carr, N, et all. (2019, October 10). Mahalo FIN7: Responding to the Criminal Operators' New Tools and Techniques. Retrieved October 11, 2019.
10. Salem, A. (2022, April 27). The chronicles of Bumblebee: The Hook, the Bee, and the Trickbot connection. Retrieved September 2, 2022.
11. Smith, S., Stafford, M. (2021, December 14). DarkWatchman: A new evolution in fileless techniques. Retrieved January 10, 2022.
12. Hod Gavriel. (2019, November 21). Dtrack: In-depth analysis of APT on a nuclear power plant. Retrieved January 20, 2021.
13. Marc-Etienne M.Léveillé. (2024, May 1). Ebury is alive but unseen. Retrieved May 21, 2024.
14. Ramin Nafisi. (2021, September 27). FoggyWeb: Targeted NOBELIUM malware leads to persistent backdoor. Retrieved October 4, 2021.
15. Quinn, J. (2019, March 25). The odd case of a Gh0stRAT variant. Retrieved July 15, 2020.
16. Symantec Security Response. (2010, January 18). The Trojan.Hydraq Incident. Retrieved February 20, 2018.
17. Lelli, A. (2010, January 11). Trojan.Hydraq. Retrieved February 20, 2018.
18. Fernando Merces, Byron Gelera, Martin Co. (2018, June 7). KillDisk Variant Hits Latin American Finance Industry. Retrieved January 12, 2021.
19. Zhang, X. (2020, February 4). Another Metamorfo Variant Targeting Customers of Financial Institutions in More Countries. Retrieved July 30, 2020.
20. Tartare, M. et al. (2020, May 21). No "Game over" for the Winnti Group. Retrieved August 24, 2020.
21. Elovitz, S. & Ahl, I. (2016, August 18). Know Your Enemy: New Financially-Motivated & Spear-Phishing Group. Retrieved February 26, 2018.
22. Nicolas Falliere, Liam O Murchu, Eric Chien 2011, February W32.Stuxnet Dossier (Version 1.4) Retrieved. 2017/09/22
23. GReAT. (2019, April 10). Project TajMahal – a sophisticated new APT framework. Retrieved October 14, 2019.
24. Black Lotus Labs. (2024, August 27). Taking The Crossroads: The Versa Director Zero-Day Exploitaiton. Retrieved August 27, 2024.