ultraviolet

CONFIDENTIAL

THREAT REPORT:

ThievingFox: There's A Fox In The Coop!

Services Performed By:
UltraViolet Cyber
Meredith Glass
(443) 351-7630
info@uvcyber.com

Published Date:
02/21/2024

Doc ID: ECORP_SUM_07032023_v1.1

Contents

1 Executive Summary

A healthy amount of focus in security is given to the prevention of initial access to sensitive information, and rightly so. However, the sheer amount of damage that can be done (financial, operational, reputational, etc.) by a threat actor post-exploitation is staggering. ThievingFox is a newly developed collection of post-exploitation tools used for gathering credentials from password managers and utilities native to the Microsoft Windows OS environment. Largely, it employs process injection and internal function hooking to achieve its purpose. While this array of tools was developed, initially, for the purpose of legal and ethical penetration testing, the code and functionality may be altered to allow threat actors to employ ThievingFox for a variety of malicious intentions. It is important for companies to have a baseline awareness of the threat this potentially poses and to know the steps that can be taken to detect, prevent and mitigate this threat.

2 Technical Analysis

ThievingFox employs functionality that assumes, at baseline, local administrator privileges have been obtained and that unfiltered SMB access is in place.

The type of process injection utilized by ThievingFox is passive, rather than active. This circumvents some of the syscall and Windows API monitoring usually expected in EDR/Antivirus and removes the need for lateral movement to achieve remote process creation. Instead, the tool uses DLL proxying in the case of the KeePassXC application in specific, replacing a targeted cryptography library (argon2.dll) with a hooked dynamic link library (dll). The masterkey to unlock a database for KeePassXC is retrieved via hook from an external function that has been handed reference to the masterkey. A heuristic is used to filter masterkeys from other data not pertinent to the tool’s functionality. In-memory byte-patching is used for the actual hook and retrieval of the masterkey is possible whenever KeePassXC unlocks a database.

COM Hijacking is employed for other native Windows applications, wherein the attacker’s malicious libraries are loaded within processes through modification of registry values. Two specific processes that are targeted are RDCMan, containing an interface possessing all necessary components for the creation of an RDP Client, and LogonUI.exe, which handles RDP connection credentials and user credentials during a session unlock by a physically present user. Notably, a fair number of credentials pass through LogonUI.exe, which means there is potential for gathering credentials from lsass.exe without manipulating or interacting with it, something that is invariably monitored via EDR.

Finally, .NET Framework applications can be targeted as well for hooking. Keepass, specifically, is an available target when utilizing ThievingFox, wherein the AppDomainManager config file is edited.

3 What UltraViolet Cyber is Doing

- UltraViolet Cyber regularly updates our IOC database to ensure alerting is up to date
- Monitoring for registry changes throughout various Windows environments, comparing changed values to what is expected to determine if malicious activity is present
- Reporting suspicious activity or access to customers, with focus maintained on role-appropriate access to data and resources

4 What Customers Can Do

- In all possible cases where it does not impact workflow or functionality, restrict administrative access to only approved users in appropriate environments and explicitly to applications/workflow requiring this type of access
- Filter SMB access, where possible and as appropriate, to the extent that it does not negatively impact functionality or restrict necessary availability of resources to those granted access
- KeePassXC automatically modifies the ACL of its process handle denying access to all users inclusive of current user, however, where KeePass is being utilized, the configuration option needs to be configured manually (Configuration/Security/ProtectProcessWithDacl)
- Regularly patch and update software and code, potentially inquiring as to whether specific applications allow for mechanisms to prevent passive injection

5 References

https://github.com/Slowerzs/ThievingFox?utm_source=tldrinfosec
https://blog.slowerzs.net/posts/thievingfox/
https://blog.quarkslab.com/post-exploitation-abusing-the-keepass-plugin-cache.html
https://gist.github.com/mgeeky/6ce72a464a691f5c105fffa1bddab301
https://www.thefinalhop.com/the-clever-craft-of-thieving-fox-a-dive-into-open-source-brilliance/
https://www.opensourceagenda.com/projects/thievingfox