DLL INJECTION ATTACK IN KERBEROS NPM PACKAGE

Dan Shallom May 15, 2020 No Comments

Written by: Dan Shallom, Cyber-security expert at OP Innovate.

TL;DR

1. There is a need for awareness of the potential risks of using open-source code
2. Introducing the *DLL preloading vulnerability* we discovered on Kerberos.
3. Mitigation & helpful tools and utilities.
4. https://www.npmjs.com/advisories/1514
5. The CVE: https://nvd.nist.gov/vuln/detail/CVE-2020-13110

For those who are not familiar with NPM (Node Package Manager), it is a gigantic software registry that contains hundreds of thousands of open source Node.js projects in the form of packages. As a matter of fact, if a developer wanted to share their code with the world, NPM would be a good way to do it.

Open Source development brings much to the world by sharing ideas, perspectives, and yes, eventually, helpful code. Unfortunately this code, in many cases, is not implemented with security in mind. Think of it like this: a machine is built to supply its clients with X, Y, and Z which it does successfully, at least at the beginning. You see this machine was brought to market without ever undergoing a rigorous battery of testing under laboratory conditions. In time defects will begin to show and it may even become dangerous to its surroundings. So in addition to diminishing reliability, it turns into a workplace liability, an accident just waiting to happen.

In general, it is not recommended to use open-source packages without fully reviewing and testing them to confirm they are

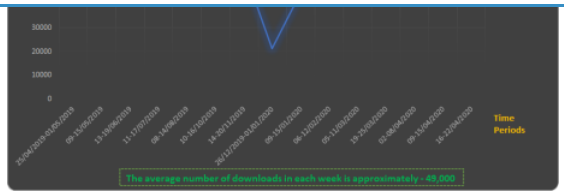


Figure 1 – Number of downloads across periods of times

Our research focused on one component used inside the package file that poses a great risk to the overall security of the user: the dynamic loading of DLLs.

The method responsible for this is called "LoadLibrary()" and it has two possible implementations: the first is supplying a fully qualified path. The second is searching for a named DLL by running through a list of directories.

Figure 2 – Unspecified path for DLLs

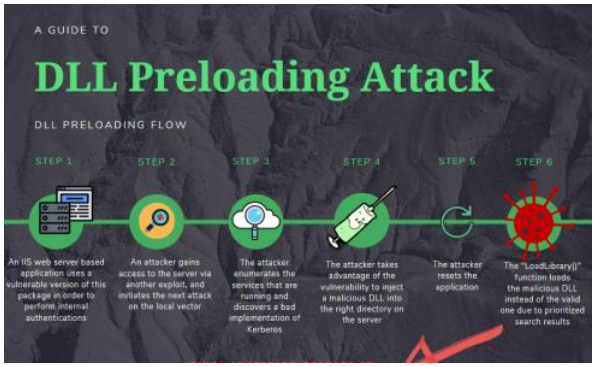
The LoadLibrary() searches for the named DLL in these directories in a set order and stops when it finds the DLL it is looking for. An attacker could therefore inject a legitimately named but malicious DLL into the first directory that is searched. The "LoadLibrary()" would find this DLL and stop its search. This malicious DLL would then be loaded into memory bringing with it potential for ACE (arbitrary code execution) and privilege escalation.

POC – Arbitrary code execution (ACE) and privilege escalation (PE)

Injecting a trojan DLL into an endpoint's memory can lead to the execution of malicious code with unlimited possibilities on the endpoint, but arguably the most interesting is privilege escalation. Using highest privileged access, an attacker gains full control over the system and can view, edit and modify system files.

The reason ACE and PE attacks are mentioned in one breath is that when they are chained together, their effect causes a severe impact on the system. When combined, the attacker can execute code remotely. Thus ACE evolves into RCE (remote code execution).

Here's how an attacker can exploit this attack:



Categories

Uncategorized

Tags

#Incident Response | API Security | Authentication | BAC | Broken Access Control | Code Review | Credentials | Cyber Breach | Cyber Security | Cyber Tips | Data Breach | DLL Injection | Equifax | Incident Response | Management | NPM | Open-Source Code | Passwords | Prototype Pollution | Ransom | Ransomware | Stay Safe

Under Cyber Attack? Click Here

Menu

- Home
- About Us
- Services
- Solutions
  - WASP
  - Ant
- Blog
- Case Studies
- Career
- Contact Us

Contact Info

Phone:
IL: +972(0)3-5771909

Email:
contactus@op-c.net

Address:
Yoqneam St 3, Tel Aviv-Yafo, 6744303, Israel.