MORPHISEC
Moving Target Defense

FACT SHEET

ASLR - What It Is and What It Isn't

Morphisec's Moving Target Defense (MTD) approach differs from ASLR. While the concepts may sound similar, ASLR lacks several key elements to make it successful at countering 0-day and targeted attacks.

Address Space Layout Randomization (ASLR) is a computer security technique which involves randomly positioning the base address of an executable and the position of libraries, heap, and stack, in a process's address space. The random mixing of memory addresses performed by ASLR means that an attack no longer knows at what address the required code (such as functions or ROP gadgets) is actually located. That way, rather than removing vulnerabilities from the system, ASLR attempts to make it more challenging to exploit existing vulnerabilities.

Main shortcomings of ASLR:

Boot-time based randomization

With ASLR, the base addresses of DLLs are based on boot-time randomization. Practically, this means that the base addresses of libraries will be randomized at the next reboot. This is an Achilles heel that can be exploited by attackers, simply by combining vulnerabilities such as memory disclosure or brute force attacks.

Unsupported executables/libraries, low-entropy

ASLR is not supported when the executable or DLLs are not built with ASLR support. Although Windows 8 and Windows 10 try to overcome this limitation (e.g., force ASLR in Windows 8), there are still exceptions that many times render the ASLR protection ineffective. Older version of Windows and legacy programs are particularly prone to this limitation. In addition, ASLR on 32-bit systems suffers from low entropy, making it vulnerable to brute force and similar attacks.

ASLR does not trap the attack

ASLR aims to prevent an attack from reliably reaching its target memory address. ASLR does not focus on trapping the attack, rather on making the attack unlikely to work. Once the shellcode jumps to the wrong address during the exploit (due to the memory randomization), the program behavior is undefined. The process might receive an exception, crash, get stuck or simply continue with inconsistent behavior as a result.

ASLR does not alert in a case of an attack

ASLR does not give any alerts about attack attempts. When a vulnerability is exploited and fails (due to ASLR's memory randomization), no alert or attack indication is received. Essentially ASLR doesn't 'know' when an attack happened.

ASLR does not provide information about an attack

Forensic information about an attack, exploitation and shellcode is crucial for any serious forensic investigation. Exploited processes, memory dumps and call stacks can be used to identify, fingerprint and tag exploits. ASLR cannot provide this information because it doesn't actually know if an attack happens or at which point it was stopped.

ASLR is being bypassed by exploits daily

Since ASLR was introduced in Windows OS, it has been bypassed countless times by real-world exploits and attacks. Attackers continuously develop new techniques to defeat ASLR defense. Bypass techniques include using ROP chain in non-ASLR modules (e.g., CVE 2013-1347), JIT/NOP spraying (e.g., CVE-2013-3346), as well as memory disclosure vulnerabilities and other techniques (e.g., CVE-2015-1685, CVE-2015-2449, CVE-2013-2556, CVE-2013-0640, CVE-2013-0634).

At Morphisec, we view ASLR as the predecessor of a multi-variable, modern Moving Target Defense approach.

Morphisec Endpoint Security delivers protection that overcomes all of the limitations listed above. In addition, our solution offers a full management system, and is able to provide rich forensic information and intelligent classification and fingerprinting of attacks for actionable remediation directives.