Linux Incident Response - Buffer Overflow and ret2libc basics

Taz Wake | Halkyn Consulting Ltd | Originally published 2023-11-07

Buffer overflow basics

A buffer overflow attack is a security vulnerability that occurs when a program writes more data to a block of memory, or buffer, than it can hold. This excess data can overflow into adjacent memory, corrupting or overwriting important data, including the program's return address, function pointers, or variables. If an attacker can control this overflow and overwrite crucial data, they can redirect the program's execution flow to malicious code they have injected into the buffer, leading to unauthorized access, privilege escalation, or even remote code execution.

What is ret2libc?

One specific type of buffer overflow attack is the ret2libc attack. This used to be very prevalent in Linux environments but is definitely less common today.

In a ret2libc attack, the attacker exploits a buffer overflow vulnerability to overwrite the program's return address with the memory address of a standard C library (libc) function. This function is then used to execute the attacker's payload. The advantage of this technique is that it allows attackers to bypass various security mechanisms, such as non-executable stacks, by utilizing existing code in the program's memory space.

During a ret2libc attack, the attacker typically crafts a payload that includes the memory address of a libc function, such as or , followed by the address of a string containing the command they want to execute (e.g., "/bin/sh"). When the vulnerable program's function returns, it jumps to the specified libc function, which, in turn, executes the attacker's desired command. This method enables attackers to achieve arbitrary code execution without injecting new code into the program, making it harder to detect and mitigate.

Investigation Tips

From a forensic perspective, detecting buffer overflow attacks and ret2libc exploits involves analyzing system and application logs, monitoring system calls, and examining memory and disk artifacts.

Unfortunately, there is no "easy button" to detect this type of attack and it often involves time-consuming reviews of disparate data sources.

Unusual patterns of memory or CPU usage, unexpected termination of processes, or unauthorized access to sensitive files can indicate a potential attack. Disk forensics may reveal unexpected changes in system binaries or the presence of unfamiliar executables, libraries, or configuration files, indicating tampering.

Additionally, analyzing network traffic for unusual patterns, such as unexpected connections or data transfers, can provide valuable clues. Intrusion detection systems (IDS) and security information and event management (SIEM) tools can assist in identifying suspicious activities, and triggering alerts when unusual patterns are detected. Regular security audits, code reviews, and penetration testing can also help identify and patch potential vulnerabilities before they are exploited, enhancing overall system security against buffer overflow attacks and similar exploits.

About Halkyn Consulting

Halkyn Consulting Ltd is a UK-based security and risk management consultancy providing 24/7 incident response, digital forensics, threat hunting, virtual security leadership (vCISO) and security compliance support to organisations worldwide. We deliver clear, practical, vendor-independent advice based on real-world threats — no hype, no upsell.

More free security resources are available at www.halkynconsulting.co.uk/security-resources.

Web: www.halkynconsulting.co.uk | **Email:** info@halkynconsulting.co.uk | **Tel:** +44 1244 940858