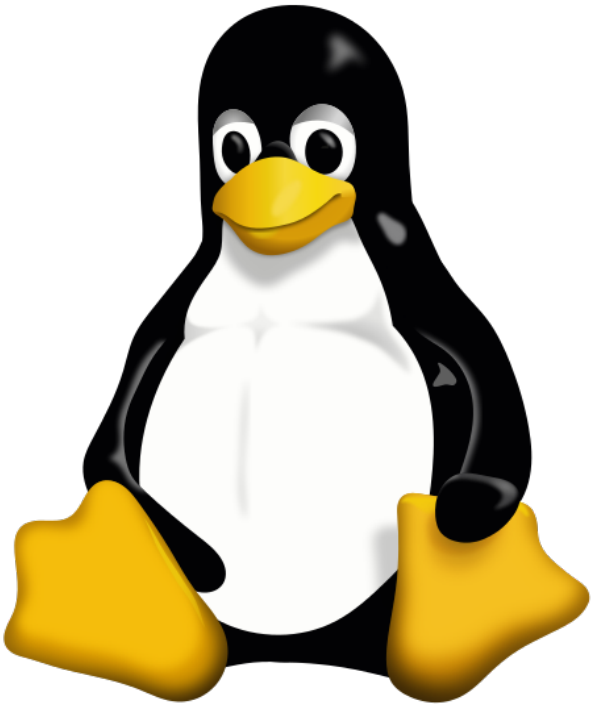
Once overlooked, uninitialized-use 'bugs' may provide portal for hacker attacks on Linux

2 March 2017, by Ben Snedeker



Popular with programmers the world over for its stability, flexibilityand security, Linux now appears to be vulnerable to hackers.

According to new Georgia Institute of Technology research, uninitialized variables - largely overlooked bugs mostly regarded as insignificant memory errors—are actually a critical attack vector that can be reliably exploited by hackers to launch privilege escalation attacks in the Linux kernel.

When successful, these intrusions give attackers increasing levels of access to a network's resources.

"While other kernel bugs and vulnerabilities have been examined and remedied, uninitialized-use bugs are not well studied, and to date, no practical defense mechanisms have been developed to protect against these attacks," said Georgia Tech Ph.D. student Kangjie Lu, lead researcher on the project.

In fact, despite potentially dangerous consequences, uninitialized-use bugs are seldom even classified as security vulnerabilities.

To prove that these bugs do present a security risk, researchers developed a novel approach, known as targeted stack spraying, to attack the operating system (OS) kernel.

Along with a technique that occupies large portions of the memory to control the stack, the automated attack probes the stack to find weaknesses that user-mode programs can exploit to direct kernel code paths and leave attacker-controlled data on the kernel stack. Ultimately, the goal of this attack is to reliably control the value of a specific uninitialized variable in the kernel space of a running program.

The research findings confirm that hackers using this method can automatically prepare a malicious pointer in the uninitialized variable. When the malicious pointer is used, a privilege escalation attack targeting the Linux kernel may occur.

"Our research shows that utilizing the targeted stack-spraying approach allows attackers to reliably control more than 91 percent of the Linux kernel stack, which, in combination with uninitialized-use vulnerabilities, suffices for a privilege escalation attack," said Lu.

Not content to merely identify the vulnerability, Lu and his fellow researchers also developed a potential solution to the problem.

"Our mitigation approach leverages the fact that uninitialized-use attacks usually control an uninitialized pointer to achieve arbitrary read/write/execution," explained Lu. "By zero-initializing pointer-type fields that the compiler cannot prove are properly initialized before they are used, we can prevent an adversary from controlling these pointers."

To limit any unnecessary performance overhead related to zero-initializing pointer-type fields, the team developed an intra-procedural program analysis that checks whether a pointer field is properly initialized when it is used. Only uninitialized pointer fields require zero initialization.

A paper titled Unleashing Use-Before-Initialization Vulnerabilities in the Linux Kernel Using Targeted Stack Spraying is being presented this week at the Network and Distributed System Security Symposium being held in San Diego, Calif.

Provided by Georgia Institute of Technology