What are Linux capabilities?

Linux capabilities break the all-or-nothing power of root into smaller units (around 40 of them) that can be assigned to an individual executable, so a program can do one privileged thing without being fully SUID-root. The risk is that some capabilities are as good as root: cap_setuid lets a program change to UID 0, cap_dac_read_search reads any file, cap_sys_admin is near-omnipotent. A binary carrying one of these, especially an interpreter, is a straightforward escalation.

HOW IT WORKS

01 The abuse and payload

The attacker lists capabilities and abuses dangerous ones:

- Enumerate: `getcap -r / 2>/dev/null`
- A Python binary with cap_setuid: `./python -c 'import os; os.setuid(0); os.system("/bin/sh")'`
- A binary with cap_dac_read_search can read protected files like `/etc/shadow` even without root.
- Perl, Ruby, and other interpreters with cap_setuid escalate the same way.

Check GTFOBins for the capability-specific one-liner. Documented techniques shown for defenders.

> THE DANGEROUS ONES
>
> *Watch for cap_setuid, cap_setgid, cap_dac_read_search, cap_dac_override, and cap_sys_admin on any binary, especially interpreters. Each is effectively a path to root or to reading any file.*

HOW TO DEFEND

- Enumerate capabilities with `getcap -r /` and remove unneeded ones: `setcap -r <file>`.
- Never grant powerful capabilities to interpreters (python, perl, ruby) or shells.
- Prefer the minimum capability that achieves the task rather than a broad one.
- Patch and review custom binaries that carry capabilities.
- Monitor for new capability assignments.

SOURCES

[1] Linux man-pages: capabilities(7)

[2] MITRE ATT&CK: Privilege Escalation (TA0004)

[3] NIST SP 800-115 Technical Guide to Security Testing

Find the privilege-escalation paths before an attacker does.

securelayer7.net/learn/privilege-escalation/what-are-linux-capabilities

Open online