What is GTFOBins?

GTFOBins is a curated, community reference that documents how legitimate Unix binaries can be abused to break out of restricted environments, escalate privileges, read or write files, or spawn a shell. For each binary it lists the exact technique under contexts like SUID, sudo, and capabilities. Attackers use it to turn a privesc finding into a working exploit instantly; defenders use the same list to know which binaries are dangerous to leave SUID or sudo-allowed.

HOW IT WORKS

01 How it is used and payload

After enumeration reveals a SUID binary or a sudo-allowed command, the attacker looks it up:

- Found a SUID `find`? GTFOBins gives: `find . -exec /bin/sh -p \; -quit`
- Allowed `sudo awk`? GTFOBins gives: `sudo awk 'BEGIN {system("/bin/sh")}'`
- A binary with `cap_setuid`? GTFOBins gives the interpreter one-liner.

It converts "this binary is privileged" into "here is the root shell." Shown for defensive context, since the same list tells defenders exactly what to lock down.

> USE IT DEFENSIVELY
>
> *Before leaving any binary SUID or adding it to sudoers, check it on GTFOBins. If it appears there, it can probably be abused to escalate, so grant it only with care or not at all.*

HOW TO DEFEND

- Cross-check every SUID binary and sudo rule against GTFOBins; remove or restrict anything that appears.
- Avoid SUID and sudo on GTFOBins-listed interpreters and tools (find, awk, python, vim, tar, and many more).
- Prefer purpose-built, minimal binaries for privileged tasks.
- Re-audit after changes, since new SUID files or sudo rules can introduce a listed binary.
- Treat a GTFOBins match as a finding, not a maybe.

SOURCES

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

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

[3] Linux man-pages: setuid(2)

Find the privilege-escalation paths before an attacker does.

securelayer7.net/learn/privilege-escalation/what-is-gtfobins

Open online