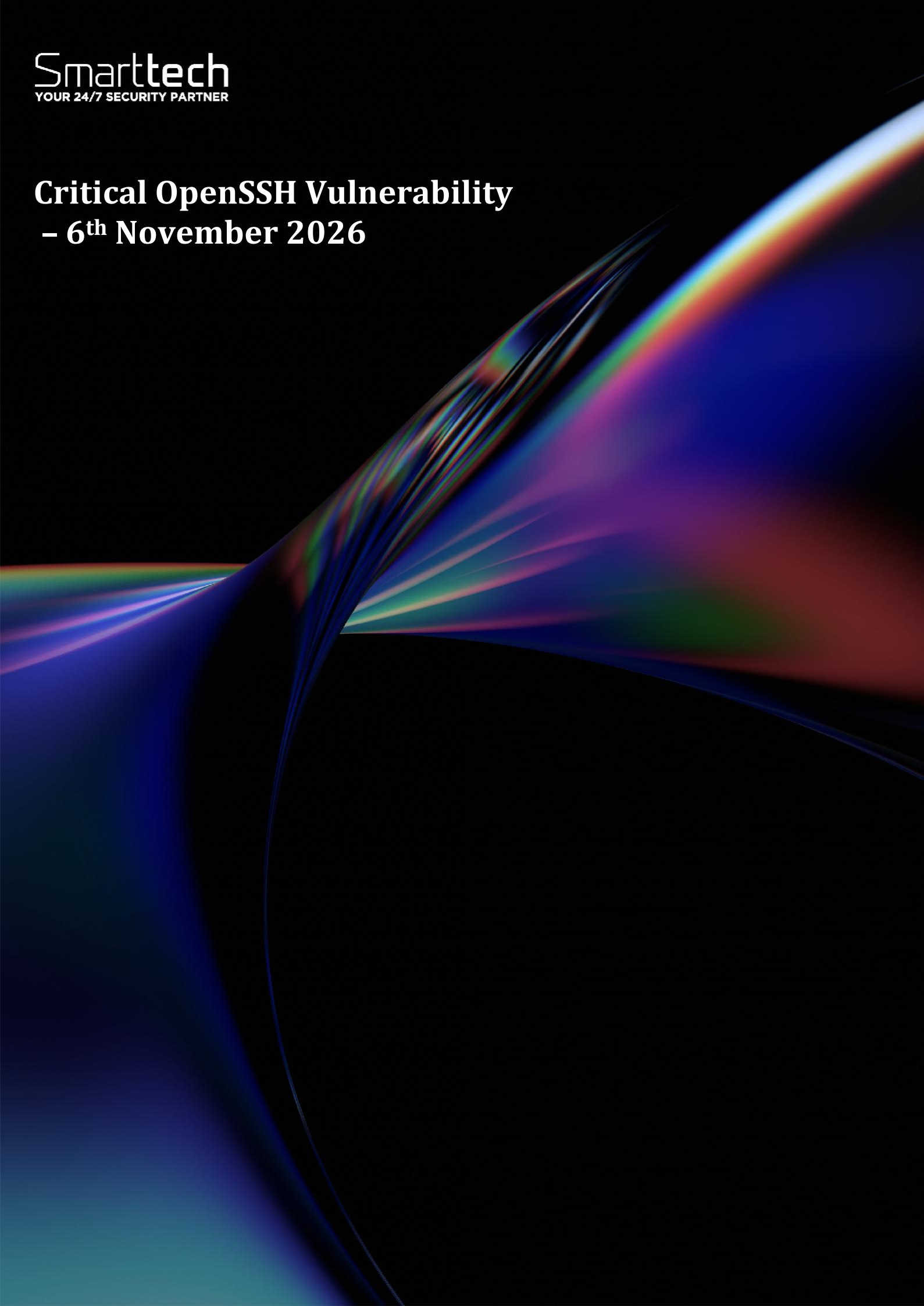
Smarttech
YOUR 24/7 SECURITY PARTNER

Critical OpenSSH Vulnerability – 6th November 2026



Document ID	SMA-Threat Report
Document status	ISSUED
Issue Number	195
Authors	Dorin Constantin Banu < constatin.banu@smarttech247.com >
Verified by	Alin Curcan < alin.curcan@smarttech247.com >
Last modified	2026-11-06
Issue Date	2026-11-05

Threat Reports are reports created by Smarttech247 based on high and critical severity vulnerabilities that may have a high potential to be exploited in the wild i.e. vulnerabilities that are present in most used products by companies and do not have an auto-update option or they are usually not automatically updated in case that could lead to some service disruption. This report is usually created as soon as the vulnerability is released, therefore we strongly recommend that the information is reviewed, tests are performed and patches are applied before the first proof-of-concept is released. Even though certain vulnerabilities may not have an active exploit in the wild at the time that we report on them, we take into consideration the wider risk and the impact it could have on systems, should an exploit like that be available after a while. Our duty is to report them on time and we recommend enterprises that, in order to keep critical business systems protected, they should consider, on average, ten working days to check whether or not the new vulnerability affects them, and if so, to implement actions in order to remove the risk.

Overview:

A newly discovered critical flaw in OpenSSH and musl, dubbed "sshole," allows attackers to log in to certain password-disabled accounts on Alpine Linux and other musl-based Linux systems by supplying an oversized password

Risk

Government:

- Large and medium government entities: **Critical**
- Small government entities: **Critical**

Businesses:

- Large and medium business entities: **Critical**
- Small business entities: **Critical**

Technical summary

A patch introduced in OpenSSH in 2016 to address the username enumeration vulnerability tracked as CVE-2016-6210 changed how accounts without valid password hashes are handled during authentication.

Linux administrators can prevent password-based logins by setting an account's password field in /etc/shadow to an asterisk (*). To prevent attackers from identifying valid usernames through timing differences, OpenSSH was modified to perform password-hashing operations even for accounts configured this way.

On systems using the musl C library without PAM authentication, the crypt() function has a 256-byte password length limit. When a password exceeds that limit, the hashing operation fails and crypt() returns * instead of a valid password hash.

Because password-disabled accounts are also represented by * in /etc/shadow, an attacker can submit an oversized password and cause OpenSSH to compare the failed hash result to the account's stored value. Since both values are *, the authentication attempt may be incorrectly treated as successful, granting access to an account that was intended to reject password logins.

The flaw does not allow attackers to bypass passwords on accounts with valid password hashes. In those cases, the * returned by crypt() does not match the account's stored password hash, causing authentication to fail.

Successful exploitation requires the following conditions:

- OpenSSH is running on a musl-based Linux system, such as Alpine Linux, OpenWrt, Buildroot, or Yocto.
- PAM authentication is disabled, as it is by default in Alpine Linux's openssh-server package.
- SSH password authentication is enabled.
- At least one account has an asterisk (*) as its password hash.
- Another account has a valid password hash.

Affected Products

OpenSSH deployments running on musl-based Linux systems, including Alpine Linux, OpenWrt, Buildroot-based systems, and Yocto-based deployments.

Recommendations

Smarttech247 team recommend the following actions be taken:

- Apply the appropriate patches or appropriate mitigations provided by OpenSSH to vulnerable systems immediately after appropriate testing.
- Upgrade software and operating systems, applications, and firmware on network assets in a timely manner for prevention.
- Use the right Vulnerability Management Tools to assess endpoint, networks or applications for known weaknesses.
- Apply the Principle of Least Privilege to all systems and services.
- Apply advanced application control and protection to enforce granular control over all application access, communications, and privilege elevation attempts.
- Kindly ensure that your Endpoint Security and Perimeter security products are updated with the latest signatures to detect these threats.

References

https://www.bleepingcomputer.com/news/security/openssh-flaw-lets-attackers-log-into-password-disabled-linux-accounts/
https://thehackernews.com/2026/11/openssh-issues-emergency-fix-for-flaw.html
https://cybersecuritynews.com/openssh-flaw-bypass-password-authentication/
https://github.com/p0dalirius/SSHole-poc

Smarttech
YOUR 24/7 SECURITY PARTNER

www.smarttech247.com