Password Recovery Services: When They Help

November 22, 2026 · Document #436 · bip39-phrase.com



Photo by Kristina Fatina on Unsplash

Learn when wallet password-recovery services can help, how success fees work, and why password guessing requires encrypted wallet data.

Contents

What a recovery service actually does 1

When recovery has a realistic chance 2

Fees, access, and the missing-file limit 2

FAQ 3

Sources 3

Example sequences from the BIP39 wordlist 3

Reading a BIP39 phrase 4

See also 5

A wallet password-recovery service can help when you still have an encrypted wallet file and remember something about its password. Its main job is to test plausible passwords against that file. It cannot reconstruct a missing wallet file from a public address, reverse a transaction, or ask a blockchain network to reset your password.

What a recovery service actually does

An encrypted wallet stores secret material behind a password. Examples include a Bitcoin Core wallet backup or an Ethereum keystore JSON file. The file contains information needed to test password candidates, including encryption parameters and, depending on the format, a salt and an integrity check.

A recovery service identifies the wallet format, checks whether the supplied data is usable, and builds a search around what you remember. Useful clues include likely words, capitalization, number sequences, approximate length, keyboard mistakes, and passwords you used around the time you created the wallet.

The service then automates candidate testing. Each candidate goes through the wallet format's password-based key derivation process. The resulting key is used to check whether decryption succeeds. Tools such as BTCRecover support password templates and typo variations for supported wallet formats.

This is targeted **password guessing**, often described as brute-forcing. Testing variations of two remembered words with a short numeric suffix is very different from searching every possible long password.

The work factor depends on both the search space and the wallet's key derivation settings. For example, Ethereum's Web3 Secret Storage format can use `scrypt` or `PBKDF2`; their parameters affect the cost of each attempt. Faster hardware helps, but it does not make an unrestricted search practical.

When recovery has a realistic chance

The strongest cases combine a readable encrypted file with specific password clues. A forgotten suffix, uncertain capitalization, or a small set of likely base passwords may produce a manageable search. A long, randomly generated password with no surviving clues usually does not.

Before contacting a provider, prepare the following:

1. Preserve the original wallet file offline and make a separate working copy. Do not let recovery software overwrite your only backup.
2. Identify the wallet application, approximate creation date, and file type. Record whether the file came from an export, a backup, or an application data directory.
3. Write down password patterns privately. Distinguish things you remember from guesses so the service can prioritize useful candidates.
4. Check for independent recovery options. A valid seed phrase, private-key backup, or another unlocked wallet installation may provide access without recovering the old password.
5. Ask what search the provider proposes, what data it needs, and what counts as success. An estimate should describe assumptions rather than promise a result.

A wallet file can also be damaged or incomplete. Password guessing cannot repair every form of corruption. Ask whether the provider has verified that the file contains the encrypted key material required for recovery.

Fees, access, and the missing-file limit

Many specialist providers use a **fee-on-success** arrangement: they receive an agreed percentage of recovered funds, or another agreed fee, if recovery succeeds. Terms vary. Some charge assessment fees, minimum fees, or separate costs for repair work, so get the complete terms in writing before sharing data.

The agreement should define success, the assets covered, how their value is calculated, who pays transaction fees, and how payment happens. A success-based fee aligns incentives, but it does not establish that a provider is trustworthy or technically competent.

Ask whether the work can use a locally generated password-verification extract instead of the complete wallet. Some formats and workflows support this; others do not. An extract may still enable offline password guessing, so treat it as sensitive. A provider's instructions should explain what the extract contains and whether it could allow spending after recovery.

Anyone who receives a complete encrypted wallet may gain spending access if they discover its password. Agree on data retention and deletion, but recognize that deletion promises cannot prove every copy disappeared. After recovery, use a trusted device to create a fresh wallet with a new seed, verify its backup, and transfer the recovered assets.

The missing-file limit follows directly from the mechanism: the service needs encrypted wallet data against which to test guesses. A public address or transaction history supplies neither the encrypted private keys nor a password verifier. Without the file, an equivalent backup, or another source of the keys, this password-recovery method has nothing to work on.

FAQ

Can a service recover my wallet using only its address?

No. An address can help identify balances and transactions, but it does not provide the encrypted wallet data needed to test passwords.

Is my wallet password the same as a BIP39 passphrase?

No. A wallet password usually encrypts local data. A BIP39 passphrase participates in deriving the seed; every passphrase produces a seed, so checking guesses requires additional information about the intended wallet.

Does a fee-on-success offer guarantee recovery?

No. It describes payment terms. Recovery still depends on usable wallet data, password clues, and a search that is computationally feasible.

Sources

- BTCRecover documentation (btcrecover.readthedocs.io)
- BIP39: Mnemonic code for generating deterministic keys (github.com)

> Continue reading on bip39-phrase.com: What Is a Seed Phrase? How a Recovery Phrase Works
> More on this topic: Recovery

Example sequences from the BIP39 wordlist

Randomly generated sequences of words from the BIP39 English wordlist. None of them passes the BIP39 checksum, so no wallet will accept them: they show the format only. Never store funds with a phrase you found in a document.

Example 1												
1	spread	2	reform	3	change	4	plug	5	say	6	echo
7	short	8	crazy	9	butter	10	tree	11	dial	12	urge

Example 2												
1	hen	2	spirit	3	deer	4	uphold	5	pass	6	hotel
7	process	8	north	9	such	10	first	11	town	12	ketchup

Example 3												
1	salon	2	admit	3	wealth	4	march	5	betray	6	glove
7	trash	8	today	9	south	10	property	11	brave	12	rely

Example 4												
1	leave	2	prize	3	rigid	4	witness	5	feature	6	remind
7	coconut	8	sausage	9	subject	10	glass	11	pepper	12	best

Example 5												
1	merit	2	muffin	3	toast	4	mix	5	learn	6	tattoo
7	course	8	keep	9	name	10	casino	11	fiscal	12	gap

Example 6												
1	toss	2	combine	3	nice	4	immune	5	double	6	flush
7	over	8	pioneer	9	swim	10	reunion	11	mechanic	12	eye

Example 7												
1	payment	2	fan	3	matter	4	fresh	5	ivory	6	caught
7	chapter	8	grit	9	lawsuit	10	disagree	11	fossil	12	alien

Example 8												
1	panic	2	black	3	remember	4	sign	5	wine	6	cute
7	heavy	8	enroll	9	clock	10	fitness	11	festival	12	finger

Example 9												
1	civil	2	chuckle	3	year	4	onion	5	device	6	donor
7	confirm	8	twin	9	draft	10	shock	11	hedgehog	12	critic

Example 10												
1	identify	2	minor	3	frozen	4	decide	5	noble	6	spike
7	ancient	8	venture	9	upset	10	loyal	11	swear	12	guess

Reading a BIP39 phrase

Every word stands for an 11-bit number: its position in the 2048-word list, counted from 0. Here is Example 1 decoded word by word. Each word links to its page on bip39-phrase.com, where the list is numbered from 1.

#	Word	Index (0-2047)	Word no.	11 bits
1	spread	1688	1689	11010011000
2	reform	1442	1443	10110100010
3	change	305	306	00100110001
4	plug	1334	1335	10100110110
5	say	1535	1536	10111111111
6	echo	559	560	01000101111
7	short	1590	1591	11000110110
8	crazy	406	407	00110010110
9	butter	250	251	00011111010
10	tree	1855	1856	11100111111
11	dial	488	489	00111101000
12	urge	1915	1916	11101111011

The 12 words carry 132 bits: 128 bits of entropy followed by a 4-bit checksum. The checksum must equal the first 4 bits of the SHA-256 hash of the entropy.

`Entropy (hex): d3168898d36bfe8bf1b1961f5cfcf477`

This sequence ends in the checksum bits 1011, but SHA-256 of its entropy starts with 1010, so wallets reject it as an invalid phrase. That is the point of the checksum: about 15 of every 16 random or mistyped sequences fail it.

See also

- Leaving Off the Last BIP39 Word: A False Trick · January 8, 2025
- Laminating a Paper Seed Backup: Help or Harm? · November 2, 2026
- Why BIP39 Words Need Only Their First Four Letters · March 30, 2025
- What a Seed Phrase Restores—and What It Leaves Out · March 31, 2025

Educational document. The example sequences are random and fail the BIP39 checksum: they are not wallets and must never hold funds. Security notes: bip39-phrase.com/security/