



ANCHOR CYBER SECURITY LLC

DLP Detection Patterns

The standard identifiers a DLP policy engine matches against — PII, payment card BIN ranges, and the secret-token formats gitleaks, trufflehog, and every major vendor scan for.

\$ dlp --pii

PII / PHI

WHAT	PATTERN	NOTE
US SSN	<code>\b(?:000 666 9\d{2})\d{3}-(?:00)\d{2}-(?:0000)\d{4}\b</code>	Excludes the reserved 000/666/9xx area numbers — a naive <code>\d{3}-\d{2}-\d{4}</code> over-matches badly.
US phone	<code>\b(?:\+?1[-.\s]?)?(?:\d{3})?[-.\s]?\d{3}[-.\s]?\d{4}\b</code>	Covers common separators; will also catch non-phone 10-digit sequences — pair with context keywords.
Email address	<code>[\w.+-]+@[\w-]+\.[\w.-]+</code>	Deliberately not RFC 5322-complete — DLP engines favor a looser, faster pattern over perfect correctness here.
Date of birth	<code>\b(0[1-9] 1[0-2])([/-])(0[1-9] 12)\d{3}[01])([/-])(19 20)\d{2}\b</code>	MM/DD/YYYY only — DOB alone usually needs a nearby keyword ("DOB," "born") to be a useful signal.

\$ dlp --pci

Payment Card (by BIN range)

NETWORK	PATTERN	NOTE
Visa	<code>4[0-9]{12}(?:[0-9]{3})?</code>	13 or 16 digits, starts with 4.
Mastercard	<code>5[1-5][0-9]{14}</code>	Older 51–55 range; the newer 2221–2720 range needs a second pattern.
Amex	<code>3[47][0-9]{13}</code>	15 digits, starts 34 or 37.
Discover	<code>6(?:011 5[0-9]{2})[0-9]{12}</code>	16 digits, starts 6011 or 65xx.

FORMAT MATCH ≠ VALID CARD

All four only confirm the number is *shaped* like that network's cards. A real DLP engine runs a Luhn checksum as a second pass to cut false positives from any 16-digit number that happens to start with a 4.

```
$ dlp --secrets
```

Credentials & Tokens

/ same signatures gitleaks / trufflehog scan for

WHAT	PATTERN	NOTE
AWS Access Key ID	AKIA[0-9A-Z]{16}	Fixed prefix, easy true-positive win in any policy.
GitHub PAT (classic)	ghp_[A-Za-z0-9]{36}	Fine-grained tokens use a different prefix (github_pat_) — add both.
Slack token	xox[baprs]-[0-9A-Za-z-]+	The letter after xox identifies token type (bot, app, etc.).
Generic high-entropy string	(?=[a-z])(?=[A-Z])(?[0-9]).{32,}	Catches unlabeled secrets no vendor prefix will match — highest false-positive rate of anything here, use as a lower-confidence signal only.

```
$ dlp --tuning
```

Tuning Notes for Policy Review

Prefix-anchored patterns (AWS, GitHub, Slack) are cheap, high-confidence, and should run first. Shape-only patterns (SSN, card numbers) need a second validation pass (checksum, or a nearby keyword) before alerting — running them unvalidated is the single fastest way to train a customer to ignore DLP alerts entirely. The high-entropy fallback belongs in a lower-priority queue, not the same alert tier as a matched AWS key.