Verifiable Credentials STRIDE Cards

This is a threat-prompt deck contextualized for the W3C Verifiable Credentials standards family and the VCALM API/lifecycle architecture. It preserves the six STRIDE and the 13-card structure per category. Each category is a violation of a security property we would like to protect.

THREAT	PROPERTY VIOLATED	THREAT DEFINITION	TYPICAL VICTIMS	EXAMPLES
Spoofing	Authentication	Pretending to be something or someone other than yourself	Processes, external entities, people	Falsely claiming to be Acme.com, winsock .dll, Barack Obama, a police officer, or the Nigerian Anti-Fraud Group
Tampering	Integrity	Modifying something on disk, on a network, or in memory	Data stores, data flows, processes	Changing a spreadsheet, the binary of an important program, or the contents of a database on disk; modifying, adding, or removing packets over a network, either local or far across the Internet, wired or wireless; changing either the data a program is using or the running program itself

Repudiation	Non-Repudiation	Claiming that you didn't do something, or were not responsible. Repudiation can be honest or false, and the key question for system designers is, what evidence do you have?	Process	Process or system: "I didn't hit the big red button" or "I didn't order that Ferrari." Note that repudiation is somewhat the odd-threat-out here; it transcends the technical nature of the other threats to the business layer.
Information Disclosure	Confidentiality	Providing information to someone not authorized to see it	Processes, data stores, data flows	The most obvious example is allowing access to files, e-mail, or databases, but information disclosure can also involve filenames ("Termination for John Doe.docx"), packets on a network, or the contents of program memory.
Denial of Service	Availability	Absorbing resources needed to provide service	Processes, data stores, data flows	A program that can be tricked into using up all its memory, a file that fills up the disk, or so many network connections that real traffic can't get through
Elevation of Privilege	Authorization	Allowing someone to do something they're not authorized to do	Process	Allowing a normal user to execute code as admin; allowing a remote person without any privileges to run code

How to use the cards

These are prompts, not controls and not normative W3C text. Use each card to ask whether the VC specification, deployment, implementation, or ecosystem flow makes the attack possible, reduces it, transfers it, accepts it, or leaves residual risk (finding a story to tell).

Random Cards Assignment

Using the order on IRC, generated by Croupier:

Player alias	Cards
Player 1	S4, S8, SK, SA, T8, TJ, R4, R8, I6, I10, D4, D9, DQ, DK, E2, E5, E7, E10, EK, EA
Player 2	S5, T2, T3, T4, TQ, TK, TA, R9, RK, RA, I4, I7, IK, D7, DA, E4, E8, E9, EJ, EQ
Player 3	S2, S6, S7, S9, S10, SJ, SQ, T9, R3, R10, RJ, I2, I3, I9, IJ, D2, D6, DJ, E6
Player 4	S3, T5, T6, T7, T10, R2, R5, R6, R7, RQ, I5, I8, IQ, IA, D3, D5, D8, D10, E3

At each round, you can choose one on the Card. You have to find a scenario and tell a story.

Cards

Spoofing

Rank	Card text
S2	An attacker can take over an endpoint normally used by a VCALM issuer, holder, verifier, status, workflow, storage, schema, context, or registry service.
S3	An attacker can try one holder binding, wallet unlock factor, access token, challenge, or exchange URL after another without being slowed down.
S4	An attacker can call an issuer, verifier, holder, status, workflow, or administration API because the VC layer assumes authentication happens elsewhere.
S5	An attacker can confuse identity by mixing issuer identifiers, DIDs, controller documents, service base URLs, instance URLs, or trust registry entries.
S6	An attacker can spoof a controller because verifiers do not check key continuity, verification method updates, service instance changes, or trust anchor changes.
S7	An attacker can act as a man in the middle because API endpoints, wallets, coordinators, or registry connections are not strongly authenticated.
S8	An attacker can steal issuer keys, verifier credentials, service tokens, OAuth client secrets, registry credentials, or signing material stored on a server.
S9	An attacker who gets a credential, wallet backup, device key, recovery phrase, or holder presentation key can act as the holder.

Rank	Card text
S10	An attacker can choose weaker authentication, proof purpose, holder binding, credential format, cryptosuite, or authorization mode and still be accepted.
SJ	An attacker can steal credentials, presentation capabilities, wallet keys, or exchange secrets stored in a browser, mobile app, extension, or client backup.
SQ	An attacker can abuse recovery, key rotation, identifier update, instance reconfiguration, or trust registry update paths to become the apparent controller.
SK	A deployment ships with demo issuer keys, default admin accounts, shared wallet secrets, sample trust anchors, broad OAuth scopes, or unchanged test credentials.
SA	You have invented a new spoofing attack against VC roles, VCALM APIs, identifiers, verification methods, instances, wallets, or trust infrastructure.

Tampering

Rank	Card text
T2	An attacker can modify a wallet SDK, issuer service, verifier service, VCALM component, resolver, context package, schema package, or cryptosuite dependency.
T3	An attacker can exploit a custom canonicalization, proof, selective disclosure, exchange, key management, or status mechanism instead of a reviewed VC mechanism.
T4	An attacker can exploit unclear boundaries between coordinator, service, storage, administration, workflow, status, and external business systems.
T5	An attacker can replay a credential, presentation, challenge, nonce, exchange URL, status response, or registry response because freshness checks are inconsistent.
T6	An attacker can write to a status list, trust list, schema, context, credential metadata, workflow definition, or verifiable data registry.
T7	An attacker can bypass policy because DIDs, URLs, media types, JSON-LD contexts, schema IDs, credential types, or issuer IDs are not canonicalized before decisions.
T8	An attacker can alter a credential, presentation, status entry, schema, context, trust response, or VCALM API message without detection.
T9	An attacker can control state such as options objects, presentation requests, workflow definitions, credential manifests, status endpoints, render methods, or schema URLs.

Rank	Card text
T10	An attacker can alter issuer records, status list entries, wallet configuration, instance configuration, trust anchors, or audit stores due to weak permissions.
TJ	An attacker can write to a vocabulary, schema, status, trust, workflow, extension, or metadata resource because access is public or over-broad.
TQ	An attacker can change challenge, domain, audience, issuer, proof purpose, status purpose, options, schema URL, or redirect target after validation.
TK	An attacker can load code through a wallet extension, credential handler, renderer, browser component, mobile extension, resolver plugin, or VCALM extension point.
TA	You have invented a new tampering attack against VC data, proofs, VCALM APIs, registries, contexts, schemas, status services, wallets, issuers, or verifiers.

Repudiation

Rank	Card text
R2	An attacker can pass credential claims, API parameters, schema terms, context values, status URLs, or presentation metadata into logs to attack log readers.
R3	A low-privilege attacker can read logs containing credential IDs, subject IDs, challenges, status checks, verifier decisions, issuer metadata, or registry lookups.
R4	An attacker can dispute a verification because the system did not record the proof, verification method, controller document version, status state, or policy used.
R5	An attacker can alter audit messages between coordinators, services, storage, status, workflow, verifier, issuer, or registry components.
R6	An attacker can trigger issuance, presentation, verification, exchange creation, key rotation, or status update events that are logged without trustworthy time.
R7	An attacker can make audit logs wrap, expire, or be overwritten so evidence of misuse, key compromise, status manipulation, or verifier abuse disappears.
R8	An attacker can make logs omit proof type, key ID, challenge, domain, audience, schema version, status version, trust list version, or validation result.
R9	An attacker can use shared issuer keys, verifier accounts, wallet operators, service accounts, or admin accounts so actions cannot be attributed.

Rank	Card text
R10	An attacker can inject unauthenticated claims, URLs, API errors, options values, exchange metadata, or presentation metadata into audit records.
RJ	An attacker can edit issuer, wallet, verifier, status, storage, workflow, or registry history without an append-only log, witness, or consistency check.
RQ	An issuer, holder, wallet, verifier, service, or registry operator can deny an issuance, presentation, verification, status update, or configuration change.
RK	A VC or VCALM component has no logs for issuance, presentation, verification, status changes, key rotation, workflow changes, or administration actions.
RA	You have invented a new repudiation attack against VC evidence, auditability, validation records, status history, registry history, or role accountability.

Information Disclosure

Rank	Card text
I2	An attacker can brute-force an encrypted wallet backup, credential archive, issuer export, verifier evidence store, or storage service because offline defenses are weak.
I3	An attacker can learn claims, credential types, schema details, issuer policy, trust decisions, status state, or rejection reasons from error messages.
I4	An attacker can read credentials or presentations because the VC or VP payload is not encrypted, even when the transport channel is encrypted.
I5	An attacker can read protected data because encryption, selective disclosure, derived presentation, or key wrapping is non-standard, weak, or incorrectly composed.
I6	An attacker can see hidden, occluded, retained, derived, rendered, or graph-connected data that the holder did not expect to disclose.
I7	An attacker can observe issuer, verifier, holder, status, schema, context, workflow, or registry requests because endpoints are not authenticated or encrypted.
I8	An attacker can read VC data through logs, caches, analytics, telemetry, search indexes, data warehouses, or debugging tools.
I9	An attacker can read wallet files, issuer configuration, verifier configuration, keys, trust lists, status sources, or exports because local permissions are weak.

Rank	Card text
I10	An attacker can read credentials, presentations, subject IDs, status indexes, schemas, trust lists, exchange records, or audit evidence from weakly protected stores.
IJ	An attacker can discover a hard-coded, shared, demo, or static key used to encrypt wallets, backups, API traffic, exports, or verifier evidence.
IQ	An attacker can read an exchange, issuance, verification, status, schema, context, or trust registry channel because it is not encrypted end to end.
IK	An attacker can learn sensitive metadata from status lookups, schema requests, context resolution, verifier queries, workflow URLs, or trust registry checks.
IA	You have invented a new information disclosure attack against VC claims, metadata, identifiers, presentations, status checks, VCALM APIs, wallets, or registries.

Denial of Service

Rank	Card text
D2	An attacker can make wallet authentication, holder binding, issuer authentication, verifier authorization, key resolution, status checking, or trust registry access unavailable.
D3	An attacker can drain a device battery by forcing a wallet to resolve contexts, fetch schemas, check status lists, verify proofs, or derive presentations repeatedly.
D4	An attacker can exhaust a secure element, HSM, or constrained key service by triggering repeated signing, proof derivation, key agreement, or verification operations.
D5	An attacker can spend the cloud budget by forcing expensive proof verification, context resolution, schema validation, status fetching, workflow fan-out, or registry checks.
D6	An unauthenticated attacker can make an issuer, verifier, holder, status, workflow, schema, context, or registry service unavailable until the attack stops.
D7	An authenticated attacker can make a wallet or client persistently unusable through malformed credentials, render methods, schemas, contexts, or cached exchanges.
D8	An authenticated attacker can make an issuer, verifier, status, storage, workflow, or registry service persistently unusable through malformed inputs or state.

Rank	Card text
D9	An unauthenticated attacker can make a wallet or client persistently unusable by causing harmful credentials, contexts, schemas, or exchange data to be stored.
D10	An unauthenticated attacker can make a service persistently unusable by filling queues, caches, storage, logs, status lists, workflows, or pending exchanges.
DJ	An attacker can stop audit, logging, monitoring, status publication, or revocation updates by flooding failed presentations, oversized credentials, or repeated status checks.
DQ	An attacker can get roughly 10-to-1 amplification by making one request trigger many context, schema, status, issuer, verifier, workflow, and registry lookups.
DK	An attacker can get roughly 100-to-1 amplification through recursive context resolution, resolver fan-out, broad status checking, or many expensive proof verifications.
DA	You have invented a new denial of service attack against VC wallets, issuers, verifiers, VCALM services, status services, schemas, contexts, registries, or proofs.

Elevation of Privilege

Rank	Card text
E2	An attacker has compromised a key supplier such as a wallet SDK, issuer SaaS, verifier SaaS, VCALM service, resolver, context host, schema host, or cryptosuite library.
E3	An attacker can access the cloud service that manages wallet keys, issuer signing keys, verifier configuration, storage services, status lists, trust anchors, or administration.
E4	An attacker can escape a wallet sandbox, browser sandbox, mobile OS sandbox, credential handler, secure element boundary, container, or verifier isolation boundary.
E5	An attacker can force data through different validation paths, such as Data Integrity vs JOSE/COSE, JSON-LD vs JWT, online vs cached status, or schema vs policy.
E6	An attacker can use permissions that are declared but not enforced, such as proof purpose, key usage, status update, credential access, exchange control, or admin rights.

Rank	Card text
E7	An attacker can pass a pointer across a trust boundary, such as a DID, URL, context, schema, statusListCredential, evidence URL, renderMethod, or workflow URL.
E8	An attacker can provide data while controlling it, have it validated, and then make it change before another component uses it with higher trust.
E9	A caller cannot tell what validation a wallet, issuer, verifier, service, resolver, registry, or status service performs before returning results.
E10	A caller cannot tell what assumptions are being made about issuers, holders, wallets, OAuth scopes, instances, key rotation, revocation freshness, or trust frameworks.
EJ	An attacker can reflect claims, display hints, evidence, render methods, API errors, status messages, or verifier text into a wallet or dashboard UI.
EQ	A privileged page, wallet view, verifier dashboard, or admin console includes user-generated credential content, render methods, evidence, images, or arbitrary URLs.
EK	An attacker can inject commands, queries, policy expressions, templates, workflow instructions, or admin actions through claims, schemas, status metadata, or presentation metadata.
EA	You have invented a new elevation of privilege attack against VC wallets, issuers, verifiers, VCALM services, resolvers, registries, renderers, or workflows.