Security Review

Vaultamagic — web-based CyberArk inventory script generator

Product	Vaultamagic (vaultamagic.com)
Type	Static web page that generates a read-only PowerShell script
What it generates	A PowerShell inventory/export script run locally by the operator
Reviewed	The web page (client-side only) and the generated PowerShell
Date	2026-06-25
Classification	Public / customer-shareable

Scope. Vaultamagic is a **static web page** that assembles a PowerShell script from values an operator enters. The page performs no authentication, makes no calls to CyberArk, and transmits nothing. All vault interaction happens later, on the operator's own machine, when they run the generated script with their own credentials. This review covers both halves: the web page and the script it produces.

1. Executive summary

Vaultamagic separates the tool from the data. The website is a pure client-side generator: it has no server, stores nothing sensitive, and never sees a credential or a vault response. It simply templates a readable PowerShell script from the operator's selections.

The generated script runs locally under the operator's own CyberArk authorization. It is read-only against the vault — it issues only GET requests plus CyberArk's read-a-secret call, and a logon/logoff. Credentials are collected at runtime via `Get-Credential`; no password is ever entered into the web page or stored in the script file. Secret retrieval is opt-in, confirmed, reason-tagged, and audited by CyberArk.

Because the page is static and self-contained (no third-party scripts, no backend), its own attack surface is minimal. The principal residual risks live in how the generated outputs are handled: plaintext secret CSVs and an optional setting that disables TLS verification. Both are detailed in section 7.

2. Trust model

The web page (client-side only)

• Static HTML, CSS, and JavaScript. No backend, no server-side processing, no database.

• No network calls to CyberArk or to any third party; the page does not even fetch external scripts or fonts.

• The only persisted state is a dark/light theme preference in `localStorage` — no vault data, no inputs, no secrets.

• Output is plain, human-readable PowerShell. The operator can review every line before running it.

The generated script (runs on the operator's machine)

• Executes in the operator's own PowerShell session with their own credentials and authorization.

• Talks only to the operator's PVWA over TLS. Writes CSV files to a local folder of the operator's choosing.

• Can do nothing the operator's CyberArk account is not already permitted to do.

3. Authentication & credentials

The script supports CyberArk, LDAP, and RADIUS logon. Credentials are gathered at runtime with `Get-Credential` and sent only in the logon request body over TLS. They are never typed into the web page and never written into the script file. **SAML/SSO is not supported** by the generated script (interactive MFA cannot be performed by a plain script); operators needing SAML should reuse a browser-obtained token or a Credential Provider integration.

4. CyberArk API calls made by the generated script

Every request the script can issue. All calls after logon carry the session token in the `Authorization` header. Only GET and POST are used.

Method	Endpoint	Purpose
POST	`/API/Auth/{CyberArk|LDAP|radius}/Logon`	Authenticate (credentials in body)
POST	`/API/Auth/Logoff`	End the session
GET	`/API/Safes`	List Safes (paginated)
GET	`/API/Safes/{id}/Members`	Safe members + permission matrix
GET	`/API/Accounts?filter=safeName eq {name}`	Accounts in a Safe (paginated)
GET	`/WebServices/PIMServices.svc/Applications`	List AAM applications
GET	`.../Applications/{appId}/Authentications`	Application auth/retrieval methods
POST	`/API/Accounts/{id}/Password/Retrieve`	Retrieve a secret (optional, opt-in)

Read-only. Apart from logon/logoff, the only non-GET call is `Password/Retrieve`, which is CyberArk's read-a-secret operation. The script issues no create, update, or delete operations.

5. Secrets handling

• Secret retrieval is **off by default**. Enabling it requires a typed YES confirmation when the script runs.

• A retrieval reason is sent with each request and recorded in CyberArk's own audit trail.

• If enabled, secrets are written to `accounts-with-secrets.csv` in plaintext (see residual risk R1). Non-secret runs contain no credential material.

6. Network & transport

• The generated script sets TLS 1.2 and talks only to the operator's PVWA.

• An optional 'Allow self-signed / private CA certificate' setting disables certificate verification for that run, to accommodate internal CAs (see residual risk R2).

• The web page itself makes no network requests at all.

7. Residual risks & recommendations

#	Risk	Recommendation	Sev.
R1	Secret export CSV is written in plaintext.	Treat the file as a secret: restrict access, store briefly, delete after migration. Avoid enabling secrets unless required.	**High**
R2	'Allow self-signed certificate' disables TLS verification for the run.	Prefer installing the enterprise CA and leaving verification on. Use the toggle only in controlled environments.	**High**
R3	Operators must trust the page's integrity before running its output.	Serve vaultamagic.com strictly over HTTPS; consider publishing a checksum of the reference script. Operators should review the script before running.	**Medium**

#	Risk	Recommendation	Sev.
R4	The script runs with the operator's full CyberArk authorization.	Run under a least-privilege account scoped to the inventory task; rely on CyberArk Safe permissions.	**Medium**
R5	Even non-secret CSVs contain sensitive metadata (Safe/account structure).	Store and share exports on a need-to-know basis.	**Low**
R6	No central audit of script generation (the page is client-side).	CyberArk audits the actual retrievals; rely on the vault's audit as the system of record.	**Low**

Vaultamagic is a read-only inventory aid for CyberArk migrations. It is not affiliated with CyberArk. Always follow your organization's credential-handling and least-privilege policies, and run the generated script from a trusted, managed endpoint.