Pretty-print ☐

{"version":3,"file":"handleAccessTokens.inline.9b374162.bundle.min.js","mappings":"mBAAO,MAgBM
A,EAAmBC,GAET,iBAAARA,EAAmBC,mBAAmBD,GAAKE,cAAcC,YAAcH,EChBhFI,EAAqB,sBAGrBC,EAAoB,IAAIC,IAC7B,
CACCF,EAJoB,iBAMpB,YACA,YACA,sBACA,eACCG,IAAKC,GAAeA,EAAWN,gBAG5BO,EAAmBC,OAAOC,yBAAyBC,SAASC,
UAAW,UAGvEC,EAA4B,CAACC,EAAiB,KACnD,GAAGX,OAAwBW,6DAAYtBC,EAAkBC,IACvB,MAAMC,EAA8B,iBAAAXD,EAAsB
A,EAAOE,MAAM,KAAK,GAAGC,OAASH,EAAOI,KAC9Eb,EAAaT,EAAAgBmBC,GAAWhB,cAE9C,OAAOG,EAAAkBiB,IAAId,IAGx
Be,EAAiBC,IACtB,IAAIC,EAOJ,OALCA,EADsB,iBAAAZD,EACUA,EAAQL,MAAM,KAAKZ,IAAKmB,GAAAMA,EAAWN,
,EAGdC,EAAkBE,OAAAQV,IAAuCD,EAAeC,KCpCllFW,EAA+B,UAC/BC,EAAsB,4BAGtB,YAAEC,EAAW,oBAAEC,EAAmb,gBA
AEC,GAAoBC,OAAOC,YAE/DC,EAAAuBH,EAEvBI,EAAAuC,CAAC,EACxCC,ED8BwC,MAC7C,MAAMC,EAAAqBC,SAAStB,OAClC
E,MAAM,KACNZ,IAAKmB,GAAAMA,EAAWN,gBAAAA,EAAAEN,QACbO,GAAAea,GAAAYgC,WAAWpC,IAAAqB,IAClEe,MAAM,KAAK,GAAId,M
ArC2B,MAC3B,MAAMJ,EAAS,UAAU0B,SAASC,YAC5BC,EAAAuB7B,IACvB8B8B,EAAe9B,EAA0BC,GAE/CN,EAAAkBoC,IAAKC,KAAKP
,SAAUI,GAAEtClC,EAAAAkBoC,IAAKC,KAAKP,SAAUK,IA4BtCG,GAAEOT,GCvCUU,GACdX,IACHD,EAZqB,kBAYSC,GA
G/B,IAAIY,EAAAgBC,MA4BpBC,iBAAAiBvB,EA1BjB,SAASwB,EAAcC,GAEtB,MAAM,OAAAEC,GAAAWD,OAAAOuB,GAAAGC,KAAK,CAAAEP,MAAAOsB,GACR,MAAMC,EAAAQ,IAAAIC,MAAM,SACxBN,EAAAOO,
MAAM,GAAAGhC,KAAAuB8B,EAAAMG,UAAAWnC,CACvDC,gBAAAiB,CAChBC,UAAAWnC,EACXC,YAAAaN,EAAAEN,QACbO,GAAAea,GAAAYgC,WA
AWpC,IAAAqB,IAClEe,MAAM,KAAK,GAAId,MAAAMf,EAAAMf,MAAAMpE,,MAAM,KAAK,IAC/B,IAAAId,GAAAmBoF,MACZzCC,CC,GAAAA8C,GACC,MAAM7D,EAAAUf
,EAAAkB4E,IAAAKvC,KAAAKAAP,UAE5CADoBhB7B,EAAAApDoBhB,MAAAMDDJpC5D,OAAA00E,eAAe7C,SAAU,SAAU,CACz
C,GAAAA8C,GACC,MAAMM7D,EAAAUf,EAAAkBk4E,IAAAKvC,KAAAPAAP,UAAA,AAAId,GAAAmBoF
,WAAAWkD,EAAAiBxF,iBAG9DO,EAAAkBoC,IAAAKvC,KAAAKAAP,UAAAPAAP,KAAP,SAAAUgD,EAAExC,EACAI,YAAY,EACZC,cAAc,cAAc,AAAc,AAAc,Ac,IAIgB,MAE/B,AE/B,EAAcN,WAAWC,YAAYT,IAAAImD,KAAKH,WAAWC,aAAEzDM,EAAAiBP,WAAWC,YAAYO,OAAAOL,KAAKH,
WAAWC,aAGrEpF,OAAO0E,eAAeS,WAAWC,YAAYZ,UAAAW,MAAO,CAC9D,GAAAA8C,GACC,MAAMM7D,EAAAUf,EAAAkB4E,IAAAKvC,KAAPAAP,UAAAPAAP
,KAAP,SAAAUgD,EAAExC,EACAI,YAAY,EACZC,cAAc,cAAc,AAAc,AAAc,Ac,KAEDEk4E,AEktrBmF,EAAAKC,OAAAeD,EAAAK,GAAAGnF,KAAAOmF,EAAAK,GAErD,IAAAIxF,EAAAeK,GAInB,OAAAO8E,EAAAKvC,KAAPAAP,UAAAPAAP,KAAP,SAAAUgD,EAAExC,EACAI,YAAY,EACZC,cAAc,cAAc,AAAc,AAAc,Ac,KAAAM,
nF,EAAAuB,IAAAhBmF,EAAAKC,OAAAeD,EAAAK,GAAAGnF,KAAAOmF,EAAAK,GErD,IAAAIxF,EAAAeK,GAErD,OAAO+E,EAAAetD,UA
AK,KAAS0D,IAAErCb,YAAY,EACZC,cAAc,KC5EhBc,E","sources":["webpack:///../../thunderbolt-
security/src/helpers.ts","webpack:///../../thunderbolt-
security/src/cookieGuard.ts","webpack:///../../thunderbolt-
security/src/handleAccessTokens.ts"],"sourcesContent":["export const isInCurrentDomain =
(clearString: string): boolean => {\n\tif (\n\t\tclearString.startsWith('//') && // Test that
it's a valid domain first, to put it into the HTTP validation rule\n\t\t/(?:[a-z0-9](?:[a-z0-
9-]{0,61}[a-z0-9])?\\.)+[a-z0-9][a-z0-9-]{0,61}[a-z0-
9]/g.test(\n\t\t\t`${location.protocol}:${clearString}`\n\t\t)\n\t) {\n\t\tclearString =
`${location.protocol}${clearString}`\n\t}\n\tif (clearString.startsWith('http')) {\n\t\t//
Test if it is the same hostname\n\t\treturn new URL(clearString).hostname ===
location.hostname\n\t}\n\treturn true\n}\n\nexport const makeStringClear = (str: string) =>
{\n\t// need to remove\n\treturn typeof str === 'string' ?
decodeURIComponent(str).toLowerCase().trimStart() : str\n}\n\nconst isCustomSetterAndGetter =
(obj: { set?: Function; get?: Function } | any) => {\n\treturn (\n\t\tobj?.set
&&\n\t\tobj?.get &&\n\t\ttypeof obj.set === 'function' &&\n\t\ttypeof obj.get === 'function'
&&\n\t\t!obj.get.toString().includes('[native code]')
&&\n\t\t!obj.set.toString().includes('[native code]')\n\t)\n}\n\n// Define the named function
for defining strict properties\nconst defineStrictProperty = (key: string, value: any, obj:
any, enumerable: boolean) => {\n\tif (isCustomSetterAndGetter(value))
{\n\t\tObject.defineProperty(obj || globalThis, key, {\n\t\t\tget: value.get,\n\t\t\tset:
value.set,\n\t\t\tconfigurable: false,\n\t\t\tenumerable: enumerable || false,\n\t\t})\n\t}
else {\n\t\tObject.defineProperty(obj || globalThis, key, {\n\t\t\tvalue,\n\t\t\twritable:
false,\n\t\t\tconfigurable: false,\n\t\t\tenumerable: enumerable ||
false,\n\t\t})\n\t}\n}\n\nexport const initOverridingGlobals = (module: any = globalThis) =>
{\n\t// Defining the script property functionality\n\tObject.defineProperty(module,
'defineStrictProperty', {\n\t\tvalue: defineStrictProperty,\n\t\twritable:
false,\n\t\tenumerable: false,\n\t\tconfigurable: false,\n\t})\n}\n","import { makeStringClear
} from './helpers'\n\nconst CLIENT_COOKIE_NAME = 'client-session-bind'\nconst CLIENT_HEADER =
'client-binding'\n\nconst deniedCookieNames = new
Set(\n\t[\n\t\tCLIENT_COOKIE_NAME,\n\t\tCLIENT_HEADER, // adding this for precaution when
hardening cookies\n\t\t'svSession',\n\t\t'smSession',\n\t\t'server-session-
bind',\n\t\t'wixSession2',\n\t].map((cookieName) => cookieName.toLowerCase())\n)\n\nconst
cookieDescriptor = Object.getOwnPropertyDescriptor(Document.prototype, 'cookie')\n\n// Returns

```
(domain: string = ``) =>\n\t ${CLIENT_COOKIE_NAME}=; ${domain} max-age=0; path=/; expires=Thu,
01 Jan 1970 00:00:01 GMT`\n\nconst removeSessionCookie = () => {\n\tconst domain =
`domain=${location.hostname};`\n\tconst cookieParamsNoDomain =
buildExpiredSessionCookie()\n\tconst cookieParams = buildExpiredSessionCookie(domain)\n\t//
This removes cookies set after we made the switch to the new model of writing cookies without
domain (for Fastly)\n\tcookieDescriptor!.set!.call(document, cookieParamsNoDomain)\n\t// This
removes cookies set with domain\n\tcookieDescriptor!.set!.call(document,
cookieParams)\n}\n\nconst isDeniedCookie = (cookie: string | { name: string }) => {\n\tconst
rawCookie = typeof cookie === 'string' ? cookie.split('=')[0].trim() : cookie.name\n\tconst
cookieName = makeStringClear(rawCookie).toLowerCase()\n\n\treturn
deniedCookieNames.has(cookieName)\n}\n\nconst filterCookies = (cookies: string | Array<{ name:
string }>): Array<string | { name: string }> => {\n\tlet normalizedCookies: Array<string | {
name: string }>\n\tif (typeof cookies === 'string') {\n\t\tnormalizedCookies =
cookies.split(';').map((_) => _.trim())\n\t} else {\n\t\tnormalizedCookies =
cookies\n\t}\n\n\treturn normalizedCookies.filter((cookie: string | { name: string }) =>
!isDeniedCookie(cookie))\n}\n\nexport const getAndDeleteClientSessionValue = () => {\n\tconst
clientSessionValue = document.cookie\n\t\t.split(';')\n\t\t.map((_) =>
_.trim())\n\t\t.filter((cookieName) => cookieName?.startsWith(CLIENT_COOKIE_NAME))
[0]\n\t\t?.split('=')[1]\n\n\tremoveSessionCookie()\n\n\treturn
clientSessionValue\n}\n\nexport const guardCookie = () => {\n\tObject.defineProperty(document,
'cookie', {\n\t\tget() {\n\t\t\tconst cookies =
cookieDescriptor!.get!.call(document)\n\t\t\tconst finalValues =
filterCookies(cookies)\n\t\t\treturn finalValues.join('; ')\n\t\t},\n\t\tset(value: string)
{\n\t\t\tconst testValue = makeStringClear(value.split(';')[0])\n\t\t\tconst shouldWrite =
[...deniedCookieNames].every(\n\t\t\t\t(deniedCookieName) =>
!testValue?.startsWith(deniedCookieName.toLowerCase()))\n\t\t\t)\n\t\t\tif (shouldWrite)
{\n\t\t\t\tcookieDescriptor!.set!.call(document, value)\n\t\t\t}\n\t\t},\n\t\tenumerable:
true,\n\t\tconfigurable: false,\n\t})\n}\n\nexport const guardCookieStore = () => {\n\t// @ts-
expect-error\n\tif (!globalThis.cookieStore) {\n\t\treturn\n\t}\n\n\t// @ts-expect-
error\n\tconst originalGet = globalThis.cookieStore.get.bind(globalThis.cookieStore)\n\t//
@ts-expect-error\n\tconst originalGetAll =
globalThis.cookieStore.getAll.bind(globalThis.cookieStore)\n\t// @ts-expect-error\n\tconst
originalSet = globalThis.cookieStore.set.bind(globalThis.cookieStore)\n\t// @ts-expect-
error\n\tconst originalDelete =
globalThis.cookieStore.delete.bind(globalThis.cookieStore)\n\n\t// @ts-expect-
error\n\tObject.defineProperty(globalThis.CookieStore.prototype, 'get', {\n\t\tvalue: async
(name: string) => {\n\t\t\tif (deniedCookieNames.has(name.toLowerCase())) {\n\t\t\t\treturn
null\n\t\t\t}\n\t\t\treturn originalGet.call(this, name)\n\t\t},\n\t\tenumerable:
true,\n\t\tconfigurable: false,\n\t})\n\n\t// @ts-expect-
error\n\tObject.defineProperty(globalThis.CookieStore.prototype, 'getAll', {\n\t\tvalue: async
() => {\n\t\t\tconst cookies = await originalGetAll.call(this)\n\t\t\treturn
filterCookies(cookies)\n\t\t},\n\t\tenumerable: true,\n\t\tconfigurable: false,\n\t})\n\n\t//
@ts-expect-error\n\tObject.defineProperty(globalThis.CookieStore.prototype, 'set',
{\n\t\tvalue: async (...args: Array<any>) => {\n\t\t\t// Handle both set('name', 'value') and
set({name, value, ...options})\n\t\t\tconst name = args.length === 1 ? args[0].name :
args[0]\n\n\t\t\tif (isDeniedCookie(name)) {\n\t\t\t\treturn\n\t\t\t}\n\n\t\t\treturn
originalSet.call(this, ...args)\n\t\t},\n\t\tenumerable: true,\n\t\tconfigurable:
false,\n\t})\n\n\t// @ts-expect-
error\n\tObject.defineProperty(globalThis.CookieStore.prototype, 'delete', {\n\t\tvalue: async
(...args: Array<any>) => {\n\t\t\t// Handle both set('name', 'value') and set({name, value,
...options})\n\t\t\tconst name = args.length === 1 ? args[0].name : args[0]\n\n\t\t\tif
(isDeniedCookie(name)) {\n\t\t\t\treturn\n\t\t\t}\n\n\t\t\treturn originalDelete.call(this,
...args)\n\t\t},\n\t\tenumerable: true,\n\t\tconfigurable: false,\n\t})\n}\n","/**\n * This
file with its imports is being loaded in the security.ejs on production using webpack.\n *
While working locally, the last bundle is being loaded in the security.ejs as minified +
uglyfied code.\n * To see changes from your code locally, after each change you need to run
\"npx webpack\" from the package folder and copy\n * the content of the generated file in
\"dist/handleAccessTokens.js\" to the security.ejs file.\n * This is only because yoshi does
let us to remove the loaded webpack-dev-server into the bundle and it causes errors locally
only.\n */\n\nimport { guardCookie, guardCookieStore, getAndDeleteClientSessionValue } from
'./cookieGuard'\n\nconst CLIENT_HEADER = 'client-binding'\nconst THUNDERBOLT_READY_EVENT_NAME
= 'tbReady'\nconst SECURITY_ERROR_TYPE = 'security_overrideGlobals'\n\n// @ts-expect-
error\nconst { experiments, siteFeaturesConfigs, accessTokensUrl } =
window.viewerModel\n\nconst accessTokensEndpoint = accessTokensUrl\nconst fetchHeaders:
```

Pretty-print

(xsrfValue) {\n\tfetchHeaders[CLIENT_HEADER] = xsrfValue\n}\n\nlet originalFetch =
fetch\n\nfunction initOnTbReady(event: Event) {\n\t// @ts-expect-error\n\tconst { logger } =
event.detail\n\ttry {\n\t\t// @ts-expect-error\n\t\twindow.tb.init({ fetch: originalFetch,
fetchHeaders })\n\t} catch (e) {\n\t\tconst error = new
Error('TB003')\n\t\tlogger.meter(`${SECURITY_ERROR_TYPE}_${error.message}`,
{\n\t\t\tparamsOverrides: {\n\t\t\t\terrorType: SECURITY_ERROR_TYPE,\n\t\t\t\teventString:
error.message,\n\t\t\t},\n\t\t})\n\n\t\t// @ts-expect-error\n\t\tif
(window?.viewerModel?.mode.debug) {\n\t\t\tconsole.error(e)\n\t\t}\n\t} finally
{\n\t\tremoveEventListener(THUNDERBOLT_READY_EVENT_NAME, initOnTbReady)\n\t\t// This is done
to remove the reference to the original fetch and use the overridden one\n\t\toriginalFetch =
fetch\n\t}\n}\n\naddEventListener(THUNDERBOLT_READY_EVENT_NAME, initOnTbReady)\n\nif
(!experiments['specs.thunderbolt.hardenFetchAndXHR']) {\n\t// @ts-expect-
error\n\twindow.fetchDynamicModel = ()
=>\n\t\tsiteFeaturesConfigs.sessionManager.isRunningInDifferentSiteContext\n\t\t\t?
Promise.resolve({})\n\t\t\t: fetch(accessTokensEndpoint, { credentials: 'same-origin',
headers: fetchHeaders }).then(function (r) {\n\t\t\t\tif (!r.ok) {\n\t\t\t\t\tthrow new
Error(`[${r.status}]${r.statusText}`)\n\t\t\t\t}\n\t\t\t\treturn r.json()\n\t\t\t})\n\t// @ts-
expect-error\n\twindow.dynamicModelPromise = window.fetchDynamicModel()\n}\n\n// Protect
reading and writing security related cookies\nguardCookie()\n\nguardCookieStore()\n"],"names":
["makeStringClear","str","decodeURIComponent","toLowerCase","trimStart","CLIENT_COOKIE_NAME","
deniedCookieNames","Set","map","cookieName","cookieDescriptor","Object","getOwnPropertyDescrip
tor","Document","prototype","buildExpiredSessionCookie","domain","isDeniedCookie","cookie","ra
wCookie","split","trim","name","has","filterCookies","cookies","normalizedCookies","_","filter
","THUNDERBOLT_READY_EVENT_NAME","SECURITY_ERROR_TYPE","experiments","siteFeaturesConfigs","ac
cessTokensUrl","window","viewerModel","accessTokensEndpoint","fetchHeaders","xsrfValue","clien
tSessionValue","document","startsWith","location","hostname","cookieParamsNoDomain","cookiePar
ams","set","call","removeSessionCookie","getAndDeleteClientSessionValue","originalFetch","fetc
h","addEventListener","initOnTbReady","event","logger","detail","tb","init","e","error","Error
","meter","message","paramsOverrides","errorType","eventString","mode","debug","console","remo
veEventListener","fetchDynamicModel","sessionManager","isRunningInDifferentSiteContext","Promi
se","resolve","credentials","headers","then","r","ok","status","statusText","json","dynamicMod
elPromise","defineProperty","get","join","value","testValue","every","deniedCookieName","enume
rable","configurable","globalThis","cookieStore","originalGet","bind","originalGetAll","getAll
","originalSet","originalDelete","delete","CookieStore","async","args","length","guardCookieSt
ore"],"sourceRoot":""}