**SUSPICIOUS COMMENT: Comment Indicates Potentially Unfinished Code -**

Line: 85 - C:\Users\Jeremy\Downloads\openssl(1).c

FIXME: Use the openssl constants instead of

**MEDIUM: Potentially Unsafe Code - assert**

Line: 499 - C:\Users\Jeremy\Downloads\openssl(1).c

The 'assert' macro usually only exists for code in the debug build. In general, no check will take place in production code. Verify that this check does not perform any critical function and is not being used in place of error handling.

assert(pkey != NULL);

**STANDARD: Potentially Unsafe Code - strlen**

Line: 609 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. For critical applications, particularly applications accepting anonymous Internet connections or unverified input data, strlen and similar functions can become victims of integer overflow or 'wraparound' errors.

if (zend\_hash\_find(Z\_ARRVAL\_P(subitem), sname, strlen(sname)+1, (void\*\*)&data) == SUCCESS) {

**STANDARD: Potentially Unsafe Code - strlen**

Line: 618 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. For critical applications, particularly applications accepting anonymous Internet connections or unverified input data, strlen and similar functions can become victims of integer overflow or 'wraparound' errors.

zend\_hash\_update(Z\_ARRVAL\_P(subitem), sname, strlen(sname)+1, &subentries, sizeof(zval\*), NULL);

**STANDARD: Potentially Unsafe Code - strlen**

Line: 626 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. For critical applications, particularly applications accepting anonymous Internet connections or unverified input data, strlen and similar functions can become victims of integer overflow or 'wraparound' errors.

zend\_hash\_update(HASH\_OF(val), key, strlen(key) + 1, (void \*)&subitem, sizeof(subitem), NULL);

**STANDARD: Potentially Unsafe Code - strlen**

Line: 657 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. For critical applications, particularly applications accepting anonymous Internet connections or unverified input data, strlen and similar functions can become victims of integer overflow or 'wraparound' errors.

if (ASN1\_STRING\_length(timestr) != strlen((char \*)ASN1\_STRING\_data(timestr))) {

**MEDIUM: Potentially Unsafe Code - getenv**

Line: 1149 - C:\Users\Jeremy\Downloads\openssl(1).c

Environment variables may be within the control of the end user and should be handled with caution. Manually check the code to ensure that the return value is checked for malicious content and is truncated, where appropriate.

config\_filename = getenv("OPENSSL\_CONF");

**MEDIUM: Potentially Unsafe Code - getenv**

Line: 1151 - C:\Users\Jeremy\Downloads\openssl(1).c

Environment variables may be within the control of the end user and should be handled with caution. Manually check the code to ensure that the return value is checked for malicious content and is truncated, where appropriate.

config\_filename = getenv("SSLEAY\_CONF");

**CRITICAL: Potentially Unsafe Code - Unsafe Use of strlcpy Allows Buffer Overflow**

Line: 1160 - C:\Users\Jeremy\Downloads\openssl(1).c

A char\* is being copied to a fixed length destination buffer and could allow a buffer overflow to take place.

strlcpy(default\_ssl\_conf\_filename, config\_filename, sizeof(default\_ssl\_conf\_filename));

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1631 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto end;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1636 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto end;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1642 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto end;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1649 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto end;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1664 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto end;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1720 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1726 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1730 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1887 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1894 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1907 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1913 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1955 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1959 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 1962 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2036 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2040 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2101 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**STANDARD: Potentially Unsafe Code - strlen**

Line: 2253 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. For critical applications, particularly applications accepting anonymous Internet connections or unverified input data, strlen and similar functions can become victims of integer overflow or 'wraparound' errors.

len = strlen(type);

**MEDIUM: Potentially Unsafe Code - memcpy**

Line: 2264 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(buffer, type, len);

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2505 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2511 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2515 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2519 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2525 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2531 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2535 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2543 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2547 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2557 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2563 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2571 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 2578 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto cleanup;

**MEDIUM: Potentially Unsafe Code - assert**

Line: 3000 - C:\Users\Jeremy\Downloads\openssl(1).c

The 'assert' macro usually only exists for code in the debug build. In general, no check will take place in production code. Verify that this check does not perform any critical function and is not being used in place of error handling.

assert(pkey != NULL);

**MEDIUM: Potentially Unsafe Code - assert**

Line: 3006 - C:\Users\Jeremy\Downloads\openssl(1).c

The 'assert' macro usually only exists for code in the debug build. In general, no check will take place in production code. Verify that this check does not perform any critical function and is not being used in place of error handling.

assert(pkey->pkey.rsa != NULL);

**MEDIUM: Potentially Unsafe Code - assert**

Line: 3018 - C:\Users\Jeremy\Downloads\openssl(1).c

The 'assert' macro usually only exists for code in the debug build. In general, no check will take place in production code. Verify that this check does not perform any critical function and is not being used in place of error handling.

assert(pkey->pkey.dsa != NULL);

**MEDIUM: Potentially Unsafe Code - assert**

Line: 3027 - C:\Users\Jeremy\Downloads\openssl(1).c

The 'assert' macro usually only exists for code in the debug build. In general, no check will take place in production code. Verify that this check does not perform any critical function and is not being used in place of error handling.

assert(pkey->pkey.dh != NULL);

**MEDIUM: Potentially Unsafe Code - assert**

Line: 3036 - C:\Users\Jeremy\Downloads\openssl(1).c

The 'assert' macro usually only exists for code in the debug build. In general, no check will take place in production code. Verify that this check does not perform any critical function and is not being used in place of error handling.

assert(pkey->pkey.ec != NULL);

**SUSPICIOUS COMMENT: Comment Indicates Potentially Unfinished Code -**

Line: 3382 - C:\Users\Jeremy\Downloads\openssl(1).c

TODO: Use the real values once the openssl constants are used

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3489 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3498 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3501 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3506 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3513 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3519 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3524 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3539 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3557 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3604 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3609 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3622 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3630 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3643 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3651 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3662 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3668 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3741 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3748 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3754 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3758 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3764 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3770 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3776 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3843 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3849 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3853 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3858 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3862 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 3868 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - memcpy**

Line: 3986 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(cryptedbuf, crypttemp, cryptedlen);

**MEDIUM: Potentially Unsafe Code - memcpy**

Line: 4110 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(cryptedbuf, crypttemp, cryptedlen);

**MEDIUM: Potentially Unsafe Code - goto**

Line: 4328 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 4338 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**MEDIUM: Potentially Unsafe Code - goto**

Line: 4354 - C:\Users\Jeremy\Downloads\openssl(1).c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto clean\_exit;

**STANDARD: Potentially Unsafe Code - strlen**

Line: 4554 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. For critical applications, particularly applications accepting anonymous Internet connections or unverified input data, strlen and similar functions can become victims of integer overflow or 'wraparound' errors.

} else if (name\_len != strlen(buf)) {

**STANDARD: Potentially Unsafe Code - strlen**

Line: 4560 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. For critical applications, particularly applications accepting anonymous Internet connections or unverified input data, strlen and similar functions can become victims of integer overflow or 'wraparound' errors.

if (!match && strlen(buf) > 3 && buf[0] == '\*' && buf[1] == '.') {

**SUSPICIOUS COMMENT: Comment Indicates Potentially Unfinished Code -**

Line: 4586 - C:\Users\Jeremy\Downloads\openssl(1).c

TODO: could expand this to make a callback into PHP user-space \*/

**MEDIUM: Potentially Unsafe Code - memcpy**

Line: 4592 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(buf, Z\_STRVAL\_PP(val), Z\_STRLEN\_PP(val)+1);

**MEDIUM: Potentially Unsafe Code - memcpy**

Line: 4824 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(iv\_new, \*piv, \*piv\_len);

**MEDIUM: Potentially Unsafe Code - memcpy**

Line: 4831 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(iv\_new, \*piv, iv\_required\_len);

**MEDIUM: Potentially Unsafe Code - memcpy**

Line: 4864 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(key, password, password\_len);

**MEDIUM: Potentially Unsafe Code - memcpy**

Line: 4961 - C:\Users\Jeremy\Downloads\openssl(1).c

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(key, password, password\_len);

**STANDARD: Potential Memory Mis-management. Variable Name: zend\_rsrc\_list\_entry rsrc TSRMLS\_DC**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: pkey**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: req->priv\_key**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: in**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: cert**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: stack**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: sk**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: p12**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: ca**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: priv\_key**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: ((req->priv\_key**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 2936 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: str**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 2936 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 3054 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: out**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 2936 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 3054 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3877 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: datain**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 2936 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 3054 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3877 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3875 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: p7**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 2936 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 3054 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3877 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3875 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3874 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: infile**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 2936 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 3054 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3877 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3875 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3874 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3805 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: outfile**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 2936 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 3054 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3877 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3875 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3874 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3805 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3806 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: sigbuf**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 2936 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 3054 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3877 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3875 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3874 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3805 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3806 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4798 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: iv**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 2936 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 3054 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3877 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3875 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3874 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3805 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3806 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4798 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4993 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: buf**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 2936 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 3054 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3877 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3875 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3874 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3805 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3806 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4798 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4993 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4453 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: ssl**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 2936 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 3054 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3877 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3875 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3874 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3805 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3806 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4798 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4993 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4453 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 4700 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: digest\_str**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 2936 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 3054 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3877 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3875 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3874 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3805 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3806 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4798 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4993 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4453 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 4700 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 4791 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

**STANDARD: Potential Memory Mis-management. Variable Name: buffer**

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 510 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4457 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2987 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

8 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3876 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

13 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3879 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1663 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 1866 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2160 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2152 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

3 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 2593 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 2936 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 3054 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3877 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3875 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

4 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3874 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3805 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 3806 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4798 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4993 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 4453 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

new without delete.

Line: 4700 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

malloc without free.

Line: 4791 FileName: C:\Users\Jeremy\Downloads\openssl(1).c

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

Line: 5100 FileName: C:\Users\Jeremy\Downloads\openssl(1).c