製品セキュリティ確保に関する基準
(STM-0117 第4版 一般用)

SONY

（注意）

この技術標準にかかる著作権をはじめとする権利は、ソニー株式会社に帰属します。

この技術標準はソニー技術マニュアル STM-0117　第 4 版の一般用です。

目　　次

1. 目的

この技術マニュアルは、設計から、量産リリースにおいて、セキュリティ視点でのルールやガイド、その他ノウハウ等を製品セキュリティ確保に関する基準として明文化したリファレンスである。この技術マニュアルの活用により各プロセスに携わる者が、製品のセキュリティ問題や脆弱性問題を予防することを目的としている。

2. 適用範囲

この技術マニュアルは、ソニー（株）が仕様を規定するソフトウェアに適用する。

3. 用語の定義

・「脆弱性」
この技術マニュアルにおける脆弱性とは製品セキュリティ上における脆弱性であり、特定の情報資産を保護するための著作権保護技術や暗号技術が破られてしまうといった問題だけでなく、悪意の第三者による人為的な攻撃や意図しない用途で利用されてしまうといった脅威の要因となる不具合や欠陥を指す。

・「バッファオーバーフロー」
プログラムが確保したバッファサイズを超えて入力データが入力された際に引き起こされる代表的な脆弱性の一つである。バッファオーバーフローによりプログラムは異常終了などの予期しない動作をするだけでなく、悪用されることによって、第三者によって作成された任意のプログラムが実行されてしまう危険性がある。

・「証明書チェイン」
証明書チェインとは、署名をほどこした認証局から信頼する認証局へと辿れる連鎖である。

・「rootkit 技術」
rootkit とは、システムにインストールされたファイルや動作するプロセスを隠すプログラムで、 OS を構成するファイルを改竄して、あるいはカーネルモードドライバーとしてインストールされる。一般には悪意ある第三者が標的のコンピューターを攻撃する際に利用するユーティリティで、侵入を手助けするバックドア機能、不正侵入中の利用記録（ログ）を抹消するデータ改竄機能、OS のモニタープログラムから攻撃プログラム（プロセス）の存在を隠蔽する機能などをもつ。

・「開発環境」
プログラムの開発で使用する、テキストエディターや make プログラム、コンパイラー、ライブラリー、SDK(Software Development Kit)など一群の開発ツールを指す。開発ツールを製品パッケージにまとめた統合開発環境（IDE)と呼ばれるものも存在する。

4. 基準

4.1 認証・身元保証

ユーザーが入力した情報を悪意ある第三者から守るために、ユーザーやその通信相手を認証し情報を保護しなければならない。この節では認証・身元保証の知識・技術について述べる。

4.1.1 https 接続時のサーバー証明書検証

WEB サービスへの接続に https 通信を用いるときは、サーバー証明書を適切に検証すること。
具体的には以下 5 点の確認が必要である。

(1) 証明書に付された認証局の署名が正しいこと
(2) 証明書チェインが正しいこと
(3) サブジェクト（証明対象の名称）が接続先と一致していること
(4) 証明書が有効期間内であること
(5) 証明書が失効していないこと（CRL の確認）

証明書の検証に失敗した場合、ユーザーに証明書の検証に失敗したことを伝え通信を終了すること。この際、証明書の検証に成功しないまま通信することの危険を伝えたうえで、通信を継続する追加の選択肢をユーザーに与えてもよい。

4.1.2 取得・表示するオンラインリソースの URL 確認手段

オンラインの情報を取得・表示する機能を有する場合は、表示中の接続先 URL を確認する手段をユーザーに提供すること。正規のサイトに接続していることを、ユーザーが確認できるようにするためである。
なお確認する手段としては、画面表示や取扱説明書記載などがある。
また WEB サーバーに接続する場合は、可能な限り https 接続を利用すること。この際には「4.1.1 https 接続時のサーバー証明書検証」の項を参照して対応すること。

4.1.3 パスワードの生表示禁止

パスワードを入力する UI は、ユーザーが入力している文字を「＊」に類する特別な記号や空白文字で表示すること。
ユーザーの入力文字そのものを時間制限なしに表示し続けてはならない。ユーザーの肩越しにパスワードを盗み見られるリスクを回避するためである。
パスワード設定済みであることを確認する UI では、ユーザーが入力した文字数とは異なる数で「＊」を表示したり、「設定済み」とだけ表示したりすることで、悪意ある第三者にパスワードの文字数のヒントを与えない工夫ができる。

4.2 製品信頼性向上

ユーザーの製品が第三者に悪用されないよう、対策しなければならない。この節では製品信頼性を向上させる知識・技術について述べる。

4.2.1 rootkit 技術の使用禁止

製品に同梱するファイルやプログラムは、セキュリティソフトウェアベンダーが提供している検出ツールで rootkit 技術が使用されていないことを必ず検査すること。rootkit として検出されるファイルやプログラムを製品に同梱してはならない。rootkit 技術の使用はユーザーに不安・不快感を与えるだけでなく、ウィルスやワームなどの隠れ場所として悪用される恐れがあるためである。

システムから実行を隠蔽するプログラムや、存在を隠すファイルやディレクトリーを作成するプログラムを、ユーザーの同意なしにインストールしてはならない。さらにユーザーの明示的な許可なしに第三者が製品を利用できるような機能を実装してはならない。

インストールプログラムでは、システムに導入するファイルやプログラムの目的を明示し、ユーザーの許諾を得てからインストールを実行すること。さらにユーザーが望んだときにインストール済みのファイルやプログラムを削除するアンインストール機能を備えること。

4.2.2 不要ポートの開放禁止

インターネットに接続する機器では、製品仕様として必要最小限のポートのみ開くこと。不要なポートを開くことで攻撃者に侵入の機会を与えるリスクを防ぐためである。

メンテナンス目的で FTP サーバーを動作させるといった場合も、出荷状態では起動しておかず、必要な場合のみ動作させるなどの工夫をすること。

外部コマンドを実行可能なコンソールを備えたOSではnetstatコマンドを使用して開いているポートを確認できる。また nmap コマンドを使うと、対象機器の外部から開いているポートを確認できる。これらのコマンドを実行した結果、製品仕様として必要不可欠と説明できないポートは必ず閉じること。

4.2.3 外部ライブラリーチェック

オープンソースを含め、外部からソフトウェアやライブラリーを調達する際は、これらが脆弱性を持たないことを確認すること。

脆弱性に対応するパッチが提供されている場合、必ず適用すること。

また、製品の出荷後に調達した外部ライブラリーに脆弱性が見つかりユーザーに被害が及び得ると判断した場合は、製品アップデータ提供やユーザーへの告知など、適切な対応をすること。

4.3 セキュリティ実装基礎技術

脆弱性防止に効果のあるコーディング時の知識・技術について述べる。

4.3.1 コーディング時の脆弱性防止に関する原則

(1) 外部データの信頼性検証

外部から与えられたデータ（通信データ、ファイル、環境変数、など）を使用する場合、想定している値の範囲や形式であるかを検証しなければならない。プログラム外部から与えられた(入力)データを、信頼できるデータであるか検証することなしに使用するべきではない。なぜなら、脆弱性問題の多くは、外部から与えられたデータが想定外の形式・値の時に、プログラムが想定外の動作をすることにより発生する。特に、このデータには人為的にプログラムを自由に操作するような悪意のあるデータが含まれていることがあるためである。

(2) バッファ長チェック

バッファを扱う処理では、必ずバッファ長を把握し、バッファ範囲を超えたアクセスが発生しないように保証しなければならない。可変長バッファを扱うときには、必ずバッファポインタとバッファサイズをセットにして扱い、バッファ範囲外を読み書きしないように範囲チェック処理を実装するべきである。なぜなら、プログラムによって割り当てられたバッファ領域を超えて書き込みが行われると、バッファオーバーフローといわれる脆弱性問題が発生し、プログラムが攻撃者に支配下されるなど最も被害の大きな事態を引き起こすことがあるためである。

(3) 使用制限されている関数の代替

(3.1) 使用制限されている関数

4.3.2、4.3.3 で定める使用制限されている関数を使用してはならない。代わりに同項 4.3.2、4.3.3 で定める代替関数の使用を推奨する。使用制限されている関数は脆弱性の原因になることの多い C 言語関数である。

(3.2) 代替関数の関数 I/F と動作仕様

4.3.2、4.3.3 で定める使用制限されている関数から代替関数への置き換えに際し、代替関数の関数 I/F および動作仕様を必ず確認し、元のプログラムの動作仕様と適合するよう実装しなければならない。使用制限されている関数と代替関数は関数 I/F および動作仕様が若干異なるためである。

(3.3) 代替関数の引数の呼び出し前検証

代替関数を呼び出す前段のプログラムにおいて、代替関数に与える引数が異常な値でないことを検証する、もしくは与える引数を正常範囲の値に補正するなどの対策を実装しなければならない。

Microsoft 環境における代替関数の多くは「パラメータの検証 (Parameter Validation)」機能を有しており、バッファオーバーフローなどの脆弱性が生じる直前でこれを検知し、プログラムを強制終了させる機能を持っている。これにより攻撃者の攻撃を失敗させることはできるが、たまたま異常な値が代替関数の引数に与えられたときにもプログラムが強制終了してしまう。

対策を実装するには、どのようなときにパラメータの検証が機能するかについて知る必要がある。MSDN ライブラリーにて各代替関数のドキュメントを参照し、パラメータの検証に関する記述を確認すること。

ただし製品仕様上、異常データを入力したときにプログラムが強制終了しても構わない場合は、代替関数に与える引数の検証・補正などの対策を省略してもよい。

(4) 最新開発環境の利用

(4.1) 新しい開発環境への移行
古い開発環境を利用している場合は、新しい開発環境への移行を検討し、可能であれば実施しなければならない。新しい開発環境ではセキュリティ強化されたライブラリーやコンパイラーを利用できるからである。

(4.2) コンパイラーやライブラリーのセキュリティ機能
コンパイラーやライブラリーに実装されたセキュリティ機能の利用を検討し、利用可能であれば活用しなければならない。このようなセキュリティ機能として、マイクロソフト C/C++ コンパイラー(cl.exe)では「ソースコード静的解析 (/analyze)」、「バッファセキュリティチェック (/GS)」、「ASLR 対応 (/DynamicBase)」、「データ実行防止互換 (/NXCompat)」、「構造化例外処理ハンドラー改竄検出 (/SafeSEH)」、GNU C/C++ コンパイラー(gcc, g++)では「バッファオーバーフロー検出(-fstack-protector)」、「整数オーバーフロー検出 (-ftrapv)」、「コンパイル時・実行時のバッファオーバーフロー検出 (-D_FORTIFY_SOURCE=2)」、「実行時のメモリー不正アクセス検出 (-g -fmudflap -lmudflap)」がある。

4.3.2 使用制限されている関数群 1

(1) 使用制限されている関数群 1 (Windows – Visual Studio 2005 以降)
対象ソフトウェアが Windows OS 上のソフトウェアであり、その開発環境が Visual Studio 2005 以降である場合、下表の使用制限されている関数を使用してはならない。対応する代替関数の使用を推奨する。

No.	使用制限されている関数	代替関数
1	gets	gets_s
2	scanf	scanf_s
3	strcpy	strcpy_s
4	strcat	strcat_s
5	sprintf	sprintf_s

(2) 使用制限されている関数群 1 (Windows – Visual Studio 2003 以前)
対象ソフトウェアが Windows OS 上のソフトウェアであり、その開発環境が Visual Studio 2003 以前である場合、下表の使用制限されている関数を使用してはならない。対応する代替関数の使用を推奨する。

No.	使用制限されている関数	代替関数
1	gets	fgets
2	scanf	sscanf
3	strcpy	strncpy
4	strcat	strncat
5	sprintf	snprintf

(3) 使用制限されている関数群 1 (Linux)
対象ソフトウェアが Linux OS 上のソフトウェアである場合、下表の使用制限されている関数を使用してはならない。対応する代替関数の使用を推奨する。

No.	使用制限されている関数	代替関数
1	gets	fgets
2	scanf	sscanf
3	strcpy	strncpy
4	strcat	strncat
5	sprintf	snprintf

(4) 使用制限されている関数群 1（その他）
対象ソフトウェアがその他の OS 上のソフトウェア、OS なしで動作するソフトウェア、あるいは OS である場合、下表の使用制限されている関数を使用してはならない。対応する代替関数の使用を推奨する。

No.	使用制限されている関数	代替関数
1	gets	fgets
2	scanf	sscanf
3	strcpy	strncpy
4	strcat	strncat
5	sprintf	snprintf

4.3.3 使用制限されている関数群 2

(1) 使用制限されている関数群 2（Windows – Visual Studio 2005 以降）
対象ソフトウェアが Windows OS 上のソフトウェアであり、その開発環境が Visual Studio 2005 以降である場合、下表の使用制限されている関数を使用してはならない。対応する代替関数の使用を推奨する。

No.	使用制限されている関数	代替関数
1	__wcserror	__wcserror_s
2	_access	_access_s
3	access	_access_s
4	_cgets	_cgets_s
5	cgets	_cgets_s
6	_cgetws	_cgetws_s
7	_chsize	_chsize_s
8	chsize	_chsize_s
9	_controlfp	_controlfp_s
10	_cprintf	_cprintf_s
11	cprintf	_cprintf_s
12	_cprintf_l	_cprintf_s_l
13	_cscanf	_cscanf_s
14	cscanf	_cscanf_s
15	_cscanf_l	_cscanf_s_l
16	_ctime32	_ctime32_s
17	_ctime64	_ctime64_s
18	_cwprintf	_cwprintf_s
19	_cwprintf_l	_cwprintf_s_l
20	_cwscanf	_cwscanf_s
21	_cwscanf_l	_cwscanf_s_l
22	_ecvt	_ecvt_s
23	ecvt	_ecvt_s
24	_fcvt	_fcvt_s
25	fcvt	_fcvt_s
26	_fprintf_l	_fprintf_s_l
27	_fscanf_l	_fscanf_s_l
28	_ftime	_ftime_s
29	_ftime32	_ftime32_s
30	_ftime64	_ftime64_s
31	_fwprintf_l	_fwprintf_s_l
32	_fwscanf_l	_fwscanf_s_l
33	_gcvt	_gcvt_s
34	gcvt	_gcvt_s
35	_getws	_getws_s
36	_gmtime32	_gmtime32_s
37	_gmtime64	_gmtime64_s
38	_i64toa	_i64toa_s
39	_i64tow	_i64tow_s
40	_itoa	_itoa_s
41	itoa	_itoa_s
42	_itow	_itow_s
43	_lfind	_lfind_s
44	lfind	_lfind_s
45	_localtime32	_localtime32_s
46	_localtime64	_localtime64_s
47	_lsearch	_lsearch_s
48	lsearch	_lsearch_s
49	_ltoa	_ltoa_s
50	ltoa	_ltoa_s
51	_ltow	_ltow_s
52	_makepath	_makepath_s
53	_mbccpy	_mbccpy_s
54	_mbccpy_l	_mbccpy_s_l
55	_mbscat	_mbscat_s
56	_mbscpy	_mbscpy_s
57	_mbslwr	_mbslwr_s
58	_mbslwr_l	_mbslwr_s_l
59	_mbsnbcat	_mbsnbcat_s
60	_mbsnbcat_l	_mbsnbcat_s_l

61	_mbsnbcpy	_mbsnbcpy_s
62	_mbsnbcpy_l	_mbsnbcpy_s_l
63	_mbsnbset	_mbsnbset_s
64	_mbsnbset_l	_mbsnbset_s_l
65	_mbsncat	_mbsncat_s *2
66	_mbsncat_l	_mbsncat_s_l *2
67	_mbsncpy	_mbsncpy_s *2
68	_mbsncpy_l	_mbsncpy_s_l *2
69	_mbsnset	_mbsnset_s
70	_mbsnset_l	_mbsnset_s_l
71	_mbsset	_mbsset_s
72	_mbsset_l	_mbsset_s_l
73	_mbstok	_mbstok_s
74	_mbstok_l	_mbstok_s_l
75	_mbstowcs_l	_mbstowcs_s_l *2
76	_mbsupr	_mbsupr_s
77	_mbsupr_l	_mbsupr_s_l
78	_mktemp	_mktemp_s
79	mktemp	_mktemp_s
80	_printf_l	_printf_s_l
81	_putenv	_putenv_s
82	putenv	_putenv_s
83	_scanf_l	_scanf_s_l
84	_searchenv	_searchenv_s
85	_snprintf	_snprintf_s *2
86	_snprintf_l	_snprintf_s_l *2
87	_snscanf	_snscanf_s
88	_snscanf_l	_snscanf_s_l
89	_snwprintf	_snwprintf_s *2
90	_snwprintf_l	_snwprintf_s_l *2
91	_snwscanf	_snwscanf_s
92	_snwscanf_l	_snwscanf_s_l
93	_creat	_sopen_s
94	_open	_sopen_s
95	_sopen	_sopen_s
96	creat	_sopen_s
97	open	_sopen_s
98	sopen	_sopen_s
99	_splitpath	_splitpath_s
100	_sprintf_l	_sprintf_s_l
101	_sscanf_l	_sscanf_s_l
102	_strdate	_strdate_s
103	_strerror	_strerror_s
104	_strlwr	_strlwr_s
105	strlwr	_strlwr_s
106	_strlwr_l	_strlwr_s_l
107	_strncat_l	_strncat_s_l *2
108	_strncpy_l	_strncpy_s_l *2
109	_strnset	_strnset_s
110	strnset	_strnset_s
111	_strnset_l	_strnset_s_l
112	_strset	_strset_s
113	strset	_strset_s
114	_strset_l	_strset_s_l
115	_strtime	_strtime_s
116	_strtok_l	_strtok_s_l
117	_strupr	_strupr_s
118	strupr	_strupr_s
119	_strupr_l	_strupr_s_l
120	_swprintf_l	_swprintf_s_l
121	_swscanf_l	_swscanf_s_l
122	_ui64toa	_ui64toa_s
123	_ui64tow	_ui64tow_s
124	_ultoa	_ultoa_s
125	ultoa	_ultoa_s
126	_ultow	_ultow_s
127	_umask	_umask_s
128	umask	_umask_s
129	_vcprintf	_vcprintf_s
130	_vcprintf_l	_vcprintf_s_l
131	_vcwprintf	_vcwprintf_s
132	_vcwprintf_l	_vcwprintf_s_l
133	_vfprintf_l	_vfprintf_s_l
134	_vfwprintf_l	_vfwprintf_s_l
135	_vprintf_l	_vprintf_s_l
136	_vsnprintf	_vsnprintf_s *2
137	_vsnprintf_l	_vsnprintf_s_l *2
138	_vsnwprintf	_vsnwprintf_s *2
139	_vsnwprintf_l	_vsnwprintf_s_l *2
140	_vsprintf_l	_vsprintf_s_l
141	__vswprintf_l	_vswprintf_s_l
142	_vswprintf_l	_vswprintf_s_l
143	_vwprintf_l	_vwprintf_s_l
144	_waccess	_waccess_s
145	_wasctime	_wasctime_s
146	_wcserror	_wcserror_s
147	_wcslwr	_wcslwr_s
148	wcslwr	_wcslwr_s
149	_wcslwr_l	_wcslwr_s_l
150	_wcsncat_l	_wcsncat_s_l *2
151	_wcsncpy_l	_wcsncpy_s_l *2
152	_wcsnset	_wcsnset_s
153	wcsnset	_wcsnset_s
154	_wcsnset_l	_wcsnset_s_l
155	_wcsset	_wcsset_s
156	wcsset	_wcsset_s
157	_wcsset_l	_wcsset_s_l
158	_wcstok_l	_wcstok_s_l

159	_wcstombs_l	_wcstombs_s_l *2
160	_wcsupr	_wcsupr_s
161	wcsupr	_wcsupr_s
162	_wcsupr_l	_wcsupr_s_l
163	_wctime	_wctime_s
164	_wctime32	_wctime32_s
165	_wctime64	_wctime64_s
166	_wctomb_l	_wctomb_s_l
167	_wfopen	_wfopen_s
168	_wfreopen	_wfreopen_s
169	_wgetenv	_wgetenv_s
170	_wmakepath	_wmakepath_s
171	_wmktemp	_wmktemp_s
172	_wprintf_l	_wprintf_s_l
173	_wputenv	_wputenv_s
174	_wscanf_l	_wscanf_s_l
175	_wsearchenv	_wsearchenv_s
176	_wcreat	_wsopen_s
177	_wopen	_wsopen_s
178	_wsopen	_wsopen_s
179	_wsplitpath	_wsplitpath_s
180	_wstrdate	_wstrdate_s
181	_wstrtime	_wstrtime_s
182	_wtmpnam	_wtmpnam_s
183	asctime	asctime_s
184	bsearch	bsearch_s
185	clearerr	clearerr_s
186	ctime	ctime_s
187	fopen	fopen_s
188	fprintf	fprintf_s
189	freopen	freopen_s
190	fscanf	fscanf_s
191	fwprintf	fwprintf_s
192	fwscanf	fwscanf_s
193	getenv	getenv_s
194	gmtime	gmtime_s
195	localtime	localtime_s
196	mbsrtowcs	mbsrtowcs_s *2
197	mbstowcs	mbstowcs_s *2
198	memcpy	memcpy_s
199	memmove	memmove_s
200	printf	printf_s
201	qsort	qsort_s
202	rand	rand_s *1
203	sscanf	sscanf_s
204	strerror	strerror_s
205	strncat	strncat_s *2
206	strncpy	strncpy_s *2
207	strtok	strtok_s
208	swprintf	swprintf_s
209	swscanf	swscanf_s
210	tmpfile	tmpfile_s
211	tmpnam	tmpnam_s
212	vfprintf	vfprintf_s
213	vfwprintf	vfwprintf_s
214	vprintf	vprintf_s
215	vsnprintf	vsnprintf_s *2
216	vsprintf	vsprintf_s
217	vswprintf	vswprintf_s
218	vwprintf	vwprintf_s
219	wcrtomb	wcrtomb_s
220	wcscat	wcscat_s
221	wcscpy	wcscpy_s
222	wcsncat	wcsncat_s *2
223	wcsncpy	wcsncpy_s *2
224	wcsrtombs	wcsrtombs_s *2
225	wcstok	wcstok_s
226	wcstombs	wcstombs_s *2
227	wctomb	wctomb_s
228	wmemcpy	wmemcpy_s
229	wmemmove	wmemmove_s
230	wprintf	wprintf_s
231	wscanf	wscanf_s

※1 rand_s()は、Windows XP 以降の Windows OS を対象とするソフトウェアでのみ使用可能。
※2 引数に_TRUNCATE を指定した場合、書き込み先バッファを超える文字列は切り捨てられる。文字列の切り捨てによりプログラムが誤動作しないよう、エラーチェックなどの注意が必要。

(2) 使用制限されている関数群 2 (Windows – Visual Studio 2003 以前)
対象ソフトウェアが Windows OS 上のソフトウェアであり、その開発環境が Visual Studio 2003 以前である場合、下表の使用制限されている関数を使用してはならない。対応する代替関数の使用を推奨する。

No.	使用制限されている関数	代替関数
1	_getws	fgetws
2	_mbscat	_mbsncat, _mbsnbcat
3	_mbscpy	_mbsncpy, _mbsnbcpy
4	swprintf	_snwprintf
5	vsprintf	_vsnprintf
6	vswprintf	_vsnwprintf
7	wcscat	wcsncat
8	wcscpy	wcsncpy

注意：VS2003 以前ではセキュリティ向上に寄与する代替関数が少ないため、表が小さくなっている。

(3) 使用制限されている関数群 2 (Linux)
対象ソフトウェアが Linux OS 上のソフトウェアである場合、下表の使用制限されている関数を使用してはならない。対応する代替関数の使用を推奨する。

No.	使用制限されている関数	代替関数
1	vsprintf	vsnprintf
2	wcscat	wcsncat
3	wcscpy	wcsncpy

注意：Linux ではセキュリティ向上に寄与する代替関数が少ないため、表が小さくなっている。

(4) 使用制限されている関数群 2 (その他)
対象ソフトウェアがその他の OS 上のソフトウェア、OS なしで動作するソフトウェア、あるいは OS である場合、下表の使用制限されている関数を使用してはならない。対応する代替関数の使用を推奨する。

No.	使用制限されている関数	代替関数
1	vsprintf	vsnprintf
2	wcscat	wcsncat
3	wcscpy	wcsncpy

4.3.4 書式文字列

製品コードでは以下の制約を遵守すること。

(1) 未検証の外部入力を書式文字列として使わない

printf/scanf ファミリーの関数の書式文字列に、プログラム外部からの入力を検証なしに渡してはならない。外部からの入力に型指定子が存在すると、プログラムをプログラマーの意図通りに制御できなくなるためである。

禁止例:

```
FILE* fp;
char buf[256];
fgets(buf, sizeof(buf), fp);
printf_s(buf); /* プログラム外部からの入力を書式文字列に与えてはならない */
```

代わりに %s 型指定をした書式に入力を与えるか、 puts で代替する。

```
printf_s("%.255s", buf);
```

(2) 型指定子 n の利用禁止

printf ファミリーの関数に型指定子 n を含む書式文字列を渡してはならない。

禁止例:

```
int nr;
printf_s("Hello,%n world!¥n", &nr); /* 型指定子 n は使用禁止 */
assert(nr == 6);
```

途中の出力文字数を利用したい場合、関数の呼び出しを分割して関数の戻り値を確認する。

```
int nr;
nr = printf_s("Hello,");
printf_s(" world!¥n");
assert(nr == 6);
```

(3) 型指定子 s による出力は精度を指定する

sprintf ファミリーの関数に型指定子 s を含む書式文字列を渡す場合、値を出力するバッファサイズに適合する精度を指定すること。

例:

```
char *key, *value;
char buf[32];
/* 32 = 15 + 1 ('=') + 15 + 1 ('¥0') */
sprintf_s(buf, sizeof(buf)/sizeof(buf[0]),
"%.15s=%.15s", key, value); /* 精度を指定 */
```

(4) 型指定子 c と s による入力は幅を指定する

scanf ファミリーの関数に型指定子 c または s を含む書式文字列を渡す場合、値を受け取るバッファサイズに適合する幅を指定すること。

例:

```
char buf[20];
sscanf_s(input, "%19s", buf, sizeof(buf)/sizeof(buf[0]));
```

4.3.5 ActiveX コントロール

信頼できないソースからの入力は使用の前に設計時に想定した範囲内の値であることを確認すること。
脆弱性問題の大半は入力値を検証し、不正な入力を許さないことで防止できる。
信頼するソースからの入力で、入力値を固定できる場合は入力値検証を省略してもよい。
なお WEB サイトを信頼する入力ソースとして扱う方法の一つとして Microsoft が提供している SiteLock ActiveX テンプレートの利用が考えられる。
配布する ActiveX コントロールにはデジタル署名を施すこと。ユーザーにコントロールの発行元の確認手段、不正な改竄がされていないことの確認手段を与えるためである。

以下のいずれの処理も行わない場合に限り Safe For Initialization と Safe For Scripting とマークしてよい。

(1) ローカルコンピューターまたはネットワーク上のファイルシステム、レジストリ設定、カメラ、USB デバイスなどに格納された情報を、任意に指定(*)して読み取り、書き込み、作成、検出、削除する
(2) 秘密鍵、パスワード、文書などのプライベート情報を開示する
(3) メモリーやディスク領域などのリソースや時間を過度に消費する
(4) ファイルの実行を含む、損害を与える可能性のあるシステムコールを実行する

Safe For Initialization/Scripting とマークしていなければ、ActiveX コントロールが不正なサイトでホスティングされたとしても、上述の 1～4 を使った悪意ある操作が実行される前に ActiveX コンテナがユーザーに警告を表示してくれる。

(*) 「任意に指定」は実行時に何らかの入力によりパスやデータの名前を操作できることを指す。なお、固定されたパス、データ名から情報を入力するときでも入力値を検証すること。

（ご注意）
ソニー技術マニュアル STM-0117 「製品セキュリティ確保に関する基準」の改定や修正に伴い、予告無しに内容を変更することがあります。

製品セキュリティ確保に関する基準
（STM-0117　第４版　一般用）

施行日：2009.04.01
発行　：ソニー株式会社
　　ソニー技術標準事務局