

radareorg / radare2 Public

<> Code Issues 773 Pull requests 46 Discussions Actions Projects

Commit 5590c87



trufae authored last week · 42 / 52 · Verified

Fix #25752 - Another command injection caused by the bad previous fix ##security

master (#25766) · 6.1.4

1 parent a8b85bc commit 5590c87

1 file changed +36 -53 lines changed

↑ Top ⚙️

Filter files...

libr/bin/format/pdb

pdb.c

1 file changed +36 -53 lines changed

Search within code ⚙️

libr/bin/format/pdb/pdb.c

```

@@ -1,4 +1,4 @@
1 - /* radare - LGPL - Copyright 2014-2025 - inisider, pancake */
1 + /* radare - LGPL - Copyright 2014-2026 - inisider, pancake */
2 2
3 3 #include <r_bin.h>
4 4
@@ -207,7 +207,7 @@ static int count_pages(int length, int page_size) {
207 207
208 208 //
209 209 static int init_pdb7_root_stream(RBinPdb *pdb, int *root_page_list, int
pages_amount, EStream indx, int root_size, int page_size) {

```

210	-	R_PDB_STREAM *pdb_stream = 0;
210	+	R_PDB_STREAM *pdb_stream = NULL;
211	211	int tmp_data_max_size = 0;
212	212	char *tmp_data = NULL, *data_end;
213	213	int stream_size = 0;
↕		@@ -324,7 +324,7 @@ static int init_pdb7_root_stream(RBinPdb *pdb, int *root_page_list, int pages_am
324	324	page->num_pages = num_pages;
325	325	} else {
326	326	page->stream_size = 0;
327	-	page->stream_pages = 0;
327	+	page->stream_pages = NULL;
328	328	page->num_pages = 0;
329	329	// R_LOG_WARN ("stream_size (%d) is 0", i);
330	330	free (tmp);
↕		@@ -356,7 +356,7 @@ static int init_pdb7_root_stream(RBinPdb *pdb, int *root_page_list, int pages_am
356	356	static void parse_pdb_info_stream(void *parsed_pdb_stream, R_STREAM_FILE *stream) {
357	357	SPDBInfoStream *tmp = (SPDBInfoStream *)parsed_pdb_stream;
358	358	
359	-	tmp->names = 0;
359	+	tmp->names = NULL;
360	360	
361	361	stream_file_read (stream, 4, (char *)&tmp->data.*version);
362	362	stream_file_read (stream, 4, (char *)&tmp->data.*time_date_stamp);
↕		@@ -395,7 +395,7 @@ static void add_index(RList *list, RBinPdb *pdb, int index, int stream_size, Est
395	395	return;
396	396	}
397	397	} else {
398	-	stream_parse_func->stream = 0;
398	+	stream_parse_func->stream = NULL;
399	399	}
400	400	r_list_append (list, stream_parse_func);
401	401	}
↕		@@ -420,7 +420,7 @@ static void fill_list_for_stream_parsing(RList *l, RBinPdb *pdb, SDbiStream *dbi
420	420	static void find_idx_in_list(RList *l, int index, SStreamParseFunc **res) {
421	421	SStreamParseFunc *stream_parse_func;

```

422 422      RListIter *it;
423 - *res = 0;
423 + *res = NULL;
424 424      r_list_foreach (l, it, stream_parse_func) {
425 425          if (index == stream_parse_func->indx) {
426 426              *res = stream_parse_func;
@@ -434,16 +434,16 @@ static int pdb_read_root(RBinPdb *pdb) {
434 434      int i = 0;
435 435      RList *pList = pdb->pdb_streams;
436 436      R_PDB7_ROOT_STREAM *root_stream = pdb->root_stream;
437 - R_PDB_STREAM *pdb_stream = 0;
438 - SPDBInfoStream *pdb_info_stream = 0;
439 - STpiStream *ss = 0;
437 + R_PDB_STREAM *pdb_stream = NULL;
438 + SPDBInfoStream *pdb_info_stream = NULL;
439 + STpiStream *ss = NULL;
440 440      R_STREAM_FILE stream_file;
441 441      RListIter *it;
442 - SPage *page = 0;
443 - SStreamParseFunc *stream_parse_func = 0;
442 + SPage *page = NULL;
443 + SStreamParseFunc *stream_parse_func = NULL;
444 444
445 445      r_list_foreach (root_stream->streams_list, it, page) {
446 - if (page->stream_pages == 0) {
446 + if (page->stream_pages == NULL) {
447 447          R_LOG_DEBUG ("no stream pages. Skipping");
448 448          r_list_append (pList, NULL);
449 449          i++;
@@ -503,25 +503,19 @@ static int pdb_read_root(RBinPdb *pdb) {
503 503
504 504      static bool pdb7_parse(RBinPdb *pdb) {
505 505          char signature[PDB7_SIGNATURE_LEN + 1];
506 - int num_root_index_pages = 0;
507 - int *root_index_pages = 0;
508 - void *root_page_data = 0;
509 - int *root_page_list = 0;
510 - int num_root_pages = 0;
511 - int num_file_pages = 0;

```

```

512 - int alloc_tbl_ptr = 0;
513 - int bytes_read = 0;
514 - int page_size = 0;
515 - int root_size = 0;
516 - int reserved = 0;
506 + int *root_index_pages = NULL;
507 + void *root_page_data = NULL;
508 + int *root_page_list = NULL;
509 +
517 510 void *p_tmp;
518 511 int i = 0;
519 512
520 - bytes_read = r_buf_read (pdb->buf, (unsigned char *)signature,
PDB7_SIGNATURE_LEN);
513 + int bytes_read = r_buf_read (pdb->buf, (unsigned char *)signature,
PDB7_SIGNATURE_LEN);
521 514 if (bytes_read != PDB7_SIGNATURE_LEN) {
522 515 // R_LOG_ERROR ("while reading PDB7_SIGNATURE");
523 516 goto error;
524 517 }
518 + int page_size, alloc_tbl_ptr, num_file_pages, reserved, root_size;
525 519 if (!read_int_var ("page_size", &page_size, pdb)) {
526 520 goto error;
527 521 }
@@ -542,12 +536,12 @@ static bool pdb7_parse(RBinPdb *pdb) {
542 536 goto error;
543 537 }
544 538
545 - num_root_pages = count_pages (root_size, page_size);
539 + int num_root_pages = count_pages (root_size, page_size);
546 540 if (num_root_pages < 1) {
547 541 R_LOG_ERROR ("Invalid page count");
548 542 goto error;
549 543 }
550 - num_root_index_pages = count_pages ((num_root_pages * 4), page_size);
544 + int num_root_index_pages = count_pages ((num_root_pages * 4), page_size);
551 545 if (num_root_index_pages > UT16_MAX) {
552 546 R_LOG_ERROR ("Invalid page count");
553 547 goto error;
@@ -625,6 +619,7 @@ static bool pdb7_parse(RBinPdb *pdb) {

```

		↑	
625	619		return false;
626	620		}
627	621		
622		+	// TODO: shouldnt this be named "fini" instead?
628	623		static void finish_pdb_parse(RBinPdb *pdb) {
629	624		if (!pdb) {
630	625		return;
		↕	@@ -638,9 +633,6 @@ static void finish_pdb_parse(RBinPdb *pdb) {
638	633		r_unref (pdb->buf);
639	634		pdb->buf = NULL;
640	635		}
641		-	
642		-	// fclose (pdb->fp);
643		-	// printf ("finish_pdb_parse()\n");
644	636		}
645	637		
646	638		static SimpleTypeMode get_simple_type_mode(PDB_SIMPLE_TYPES type) {
		↕	@@ -1384,10 +1376,10 @@ static void print_types(const RBinPdb *pdb, PJ
		↑	*pj, const int mode) {
1384	1376		
1385	1377		////////////////////////////////////
			//
1386	1378		static void print_gvars(RBinPdb *pdb, ut64 img_base, PJ *pj, int format) {
1387		-	SStreamParseFunc *omap = 0, *sctns = 0, *sctns_orig = 0, *gsym = 0, *tmp;
1388		-	SIMAGE_SECTION_HEADER *sctn_header = 0;
1389		-	SGDATAStream *gsym_data_stream = 0;
1390		-	SPEStream *pe_stream = 0;
1379		+	SStreamParseFunc *omap = NULL, *sctns = NULL, *sctns_orig = NULL, *gsym =
			NULL, *tmp;
1380		+	SIMAGE_SECTION_HEADER *sctn_header = NULL;
1381		+	SGDATAStream *gsym_data_stream = NULL;
1382		+	SPEStream *pe_stream = NULL;
1391	1383		SGlobal *gdata;
1392	1384		RListIter *it;
1393	1385		char *name;
		↕	@@ -1433,43 +1425,34 @@ static void print_gvars(RBinPdb *pdb, ut64
		↑	img_base, PJ *pj, int format) {
1433	1425		sctn_header = r_list_get_n (pe_stream->sections_hdrs, (gdata->segment
			- 1));

```

1434 1426         if (sctn_header) {
1435 1427             char *filtered_name;
1428 +             char sname[PDB_SIZEOF_SECTION_NAME + 1];
1429 +             r_str_ncpy (sname, sctn_header->name, sizeof (sname));
1430 +             r_str_sanitize (sname);
1436 1431             name = r_bin_demangle_msvc (gdata->name.name);
1437 1432             name = (name)? name: strdup (gdata->name.name);
1433 +             ut64 addr = img_base + omap_remap ((omap)? (omap->stream): 0,
gdata->offset + sctn_header->virtual_address);
1438 1434             switch (format) {
1439 1435                 case 2:
1440 1436                 case 'j': // JSON
1441 1437                     pj_o (pj);
1442 1438                     pj_kN (pj, "address", (img_base + omap_remap ((omap)? (omap-
>stream): 0, gdata->offset + sctn_header->virtual_address)));
1443 1439                     pj_kN (pj, "symtype", gdata->symtype);
1444 -                     pj_ks (pj, "section_name", sctn_header->name);
1440 +                     pj_ks (pj, "section_name", sname);
1445 1441                     pj_ks (pj, "gdata_name", name);
1446 1442                     pj_end (pj);
1447 1443                     break;
1448 1444                 case 1:
1449 1445                 case '*':
1450 -                 case 'r':
1446 +                 case 'r': // r2 script
1451 1447                     filtered_name = r_name_filter_dup (r_str_trim_head_ro
(name));
1452 -                     pdb->cb_printf ("f pdb.%s = 0x%" PFMT64x " # %d %.*s\n",
1453 -                     filtered_name,
1454 -                     (ut64) (img_base + omap_remap ((omap)? (omap->stream): 0,
gdata->offset + sctn_header->virtual_address)),
1455 -                     gdata->symtype,
1456 -                     PDB_SIZEOF_SECTION_NAME,
1457 -                     sctn_header->name);
1458 -                     char *b64name = r_base64_encode_dyn ((const ut8 *)name,
strlen (name));
1459 -                     if (b64name) {
1460 -                         pdb->cb_printf ("fN pdb.%s base64:%s\n", filtered_name,
b64name);
1461 -                         free (b64name);

```

1462	-	}
1448	+	pdb->cb_printf ("'"@0x%" PFMT64x "'f pdb.%s\n", filtered_name);
1463	1449	free (filtered_name);
1464	1450	break;
1465	-	case 'd':
1451	+	// case 'd':
1466	1452	default:
1467	-	pdb->cb_printf ("0x%08" PFMT64x " %d %.*s %s\n",
1453	+	pdb->cb_printf ("0x%08" PFMT64x " %d %s %s\n",
1468	1454	(ut64) (img_base + omap_remap ((omap)? (omap->stream): 0, gdata->offset + sctn_header->virtual_address)),
1469	-	gdata->symtype,
1470	-	PDB_SIZEOF_SECTION_NAME,
1471	-	sctn_header->name,
1472	-	name);
1455	+	gdata->symtype, sname, name);
1473	1456	break;
1474	1457	}
1475	1458	free (name);
⌵		@@ -1516,7 +1499,7 @@ R_API bool r_bin_pdb_parser_with_buf(RBinPdb *pdb, R_OWNED RBuffer *buf) {
1516	1499	pdb->pdb_parse = pdb7_parse;
1517	1500	R_FREE (signature);
1518	1501	pdb->pdb_streams = r_list_new ();
1519	-	pdb->stream_map = 0;
1502	+	pdb->stream_map = NULL;
1520	1503	pdb->finish_pdb_parse = finish_pdb_parse;
1521	1504	pdb->print_types = print_types;
1522	1505	pdb->print_gvars = print_gvars;
⌵		

Comments 0



Please [sign in](#) to comment.