Appendix Data-driven Mutation Testing:

GomSpace:

LibGCSP Fault Model

Target function

The probes will be inserted in the function *csp_service_handler*, contained in the file *libgscsp/lib/libcsp/src/csp_service_handler.c*
This function handles an incoming connection and depending on the destination port, handles the packet to the corresponding service handler. The function takes as a parameter a connection structure of type *csp_conn_t* and a CSP packet of type *csp_packet_t*.

Target data structure:

In this section, we describe the data structures targeted for the data-driven mutation process.

csp_conn_t

Member	**Type**
type	csp_conn_type_t
state	csp_conn_state_t
idin	csp_id_t (union)
idout	csp_id_t (union)
rx_event	csp_queue_handle_t
rx_queue	csp_queue_handle_t
socket	csp_queue_handle_t
timestamp	uint32_t
opts	uint32_t
rdp	csp_rdp_t

We are interested in mutating *idin* field, which contain the header of the CSP packet on the incoming connection.

csp_id_t

The packet header is represented by the data structure *csp_id_t* and contains the following fields.

Member	**Type**	**Description**
ext	uint32_t	
pri	unsigned int	Priority

src	unsigned int	Source
dst	unsigned int	Destination
dport	unsigned int	Destination port
sport	unsigned int	Source port
flags	unsigned int	Flags

Identifier Fault Model

The probe will be inserted before the switch statement, before the invocation of the service handlers.

Member	**Type**	**Fault Classes**
ext	uint32_t	None
pri	unsigned int	INV(Min=0; Max=3; D=0), VAT(Threshold=3; Delta=1), BF(Min=0; Max=0; State=1), BF(Min=1; Max=1; State=1),
src	unsigned int	INV(Min=0; Max=31; D=0), VAT(Threshold=31;D=1)
dst	unsigned int	INV(Min=0; Max=31; D=0), VAT(Threshold=31;D=1)
dport	unsigned int	INV(Min=8; Max=30; D=0), VOR(Min=8; Max=30;D=1), SS(Delta=1), SS(Delta=-1)
sport	unsigned int	INV(Min=8; Max=30; D=0), VOR(Min=8; Max=30;D=1), SS(Delta=1), SS(Delta=-1)
flags	unsigned int	BF(Min=0; Max=0; State=1), BF(Min=1; Max=1; State=1), BF(Min=2; Max=2; State=1), BF(Min=3; Max=3; State=1)

One of the objectives of the fault model is to mutate the *dport* field from the identifier; a different value of this field changes the service required. In the following, a table with the full list of services is presented.

Service	**Code**	**Description**
GS_CSP_CMP	0	CSP Management Protocol
GS_CSP_PING	1	CSP Ping
GS_CSP_PS	2	Process status
GS_CSP_MEM_FREE	3	Show memory free
GS_CSP_REBOOT	4	Reboot/reset request
GS_CSP_BUF_FREE	5	Number of free CSP buffers
GS_CSP_UPTIME	6	Show uptime
GS_CSP_PORT_RPARAM	7	Parameter service (libparam)
GS_CSP_PORT_FTP	9	File Transfer Service (libftp)

GS_CSP_PORT_RLOG	11	Remote log service (liblog)
GS_CSP_PORT_RGOSH	12	Remote GOSH service (librgosh)
GS_CSP_PORT_AIS	13	AIS command port (libais)
GS_CSP_PORT_ADSB	14	ADS-B command port (libadsb)
GS_CSP_PORT_GSSB	16	GomSpace Sensor Bus (libgssb)
GS_CSP_PORT_FP	18	Flight Planner (libfp)
GS_CSP_PORT_ADCS	20	ADCS (libadcs)
GS_CSP_PORT_HK	21	House Keeping (libhk)
GS_CSP_PORT_GSCRIPT	22	G(omSpace) script service (libgosh)
GS_CSP_PORT_MONITOR	26	Gomspace Monitor application command port
GS_CSP_PORT_REMOTE_SHELL	27	Remote shell (libgosh)
GS_CSP_PORT_HK_BEACON	30	Housekeeping beacon port (libhk)