

TRUSTINSOFT ANALYZER FOR C/C++

TrustInSoft Analyzer (TISA) delivers exhaustive static and dynamic analysis for C and C++ code, ensuring software safety, security, and reliability through advanced formal methods like abstract interpretation. TISA supports compliance with key industry standards while integrating seamlessly into existing workflows, providing mathematically proven assurance.





)KEY FEATURES



Exhaustive Analysis

Detects all undefined behaviors, including buffer overflows, integer overflows, and division by zero, with zero false alarms.



Full Path Context Sensivity

Thorough exploration of all execution paths, ensuring comprehensive coverage and robust code integrity.



Automated Generalization

Tests all potential code values, significantly increasing test coverage and reducing testing effort.



Hardware-Aware Analysis

Simulate and test software on specific hardware targets (X86, ARM, RISC-V) with detailed memory mapping.



Advanced Reporting Tools

Streamlined generation of compliance reports, ensuring adherence to industry standards.



Seamless CI/CD Integration

Easily integrates into Agile and V-model workflows, enhancing your verification processes without disrupting existing pipelines.

STANDARDS COMPLIANCE

- ISO 26262
- ISO 21434
- DO -178C
- IEC 61508
- AUTOSAR
- CERT C

AUTOMATED REPORTS

INVESTIGATION TOOLS

Tube Summary of provement Tube Summary of provement Tube Tube Tube<											
Summary of verweinter Notice of the state of	Trustle Soft Analyzes Depart								The processing of the end of the second se		
Notice with the state with the stat								-			
Notice with the state with the stat	Created on 11/17/2022						nmary o	fallyzer version 1.44	4 int shealute int(long x)		
Individual Manages Image:	Analysis Summary Consolidated Alarms Individual Analyses Une Coverage Statement Coverage MC/DC Coverage Report Metadata								Run: Parsing Value Analysis 🕲 🗸		
with a way set 	(lests		No configuration file found Learn more about configuration files.		
vis. syn Ausyles Function Av Statement Av Table av Manages Av 0 Location-sht 00% <td colspan="6">Individual Analyses</td> <td></td> <td></td> <td></td>	Individual Analyses										
1 postore-white 00% 00% 00% 00% 00% 00% 00% 00% 00% 00	#UBs ▲/▼	Analysis 🔺	Function Coverage	√▼ Statement Coverage	▲/▼ Semantic Statement Coverage	▲/▼	Total ▲/▼	Memory ▲/▼	if (x < (long)8) {		
D 2.regarder with 60% 99% 87% 2.77% 0.05%es digner with 60% 99% 87% 2.77% 0.05%es digner with 60% 59% 80% 2.780 s 0.05%es digner with 60% 51% 80% 6.1% s 0.05%es digner with 60% 6.1% s digner wit	0	1.positive-shift			-			•	Signed overflow		
Summarized on is 59% 96% 2200 5 0 Geyres Single description of the second descriptio		2 accessible adulta	600	50%	070		0.771 -	0.05.444	✓ 2 files successfully compiled. Predicate		
	v		60%	39%	0776		2.7718	0 Gbytes			
Ageneralized: 00% 01% 01% 0.11% 0 objects Minimum 20% 01% 01% 0.11% 0 objects Minimum 20% 01% 01% 0.11% 0 objects Minimum 20% 01% 01% 01% 01% 01% 01% 01% 01% 01% 0	1	3.generalized- shift	60%	59%	96%		2.780 s	0 Gbytes			
<pre>but sting but bits to be an an and the set of the</pre>		4.oeneralized-									
<pre>cverifier in the state is in the state is</pre>	v	string	60%	01%	100%		5.113 8	0 Gbytes			
Control who abolate_int Casase 1 Signeralized-shift Image: abolate_int(ing s) Bug Control mark Image: abolate_int(ing s) Bug Control mark Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s) Image: abolate_int(ing s)	Kind	▲ Functio	on ▲/	v File ▲	/v Occurrences	▲/▼	Analyses				
Image: intervention destines and constrained in the destines intervention destines and constrained intervente and constrained intervention destines and c	Overflow	absolu	ite int	caesar c	1			ft			
Bug Location *** 92337283654775807 *** 92337283654775807 *** 92337283654775807 *** 92337283654775807 *** 92337283654775807 *** 92337283654775807 *** 92337283654775807 *** 92337283654775807 *** 92337283654775807 *** 92337283654775808 *** 1000 umplift **** 1000 umplift **** 1000 umplift ************************************											
Bug Location Precise rest coverage Display Test Coverage Display Extensive investigation tools: Control flow and catasets via table to be: Control fl	-1 int	-1 int absolute.int(long x)							Analysis / network (int)ake vi		
<pre>if (% 0)</pre>	1	Bug Location							Summary nation nation		
<pre></pre>	3								Current Function: absolute and Statistical y		
<pre>predicate</pre>	5								m_enuceds		
Preciate +* Preciate +* \$ 22337283684775807 Values and expression * (- 223372836847588811) <td>7</td> <td></td> <td colspan="3"></td>	7										
Prediction in x to 22337238654775887 Values and expression ix to 1-2233723865477588711 ix to 1-2233723865477588711 ix to 1-2233723865477588811 ix to 1-22337238654775888128 ix to 1-22337238654775888138 ix to 1-22347888 ix to 1-223478888 ix to 1-223478888	9) 9										
	Predicate										
Values and expression * c [-2223228086545757888.,-1] * c [-2223228086545757888.,-1] * c [-2223228086545757888.,-1] * c [-22232808654757888.,-1] * c [-2223280865477888.,-1] * c [-222380878 * c [-22											
Springer 2000			567				Slevel: / 1000 Merge after / Enabled				
** (2233220366567378881) ** (223322036567378881) ** (223322036567378881) ** (223322036567378881) ** (223322036567378881) ** (223322036567378881) ** (223322036567378881) ** (223322036567378881) ** (223322036567378881) ** (223322036567378881) ** (2233220365678881) ** (2233220365678881) ** (223328678881) ** (223328678881) ** (223328678881) ** (223328678881) ** (22332868678881) ** (2233286878881) ** (2233286878881) ** (22332888881) ** (2233288881) ** (22332888881) ** (22332888881) ** (22332888881) ** (22332888881) ** (22332888881) ** (223328888881) ** (22332888881) ** (22332888881) ** (22332888881) ** (22388881)	Values and expression										
Image: Construction of the constrult of the construction of the constructio	x ∈ [-92233720368547758081]							KL Function (caliste) Status Before			
Get (f) (f	ferfande 4 milit andre 4 milit andre 4 milit andre 5 militaria e an antibilitativa e antibilitativa e antibilitativa										
G (() (G = C(Chaldrent - 1) B) () () () () () () () () (66 - (Caller caesar_decrypt Both No term selected		
Test Coverage Display	68 if (s (TCSha296State_t) 0								Caller caesar_encrypt Both		
Test Coverage Display Cateros	70 return TC_ORYPTO_FAIL:										
73 with (detailer > 0) (75 with (detailer > 0) (76 x) (detailer) = (detailer) 77 x) (-interesting action tools: Control flow and control flow (call stack variable values)	72	72 return TC_OVPT0_SUCCESS; Test Coverage Display							Extensive investigation tools: Control flow and		
a solutions solutions of the constraints of the con	74	74									
udid IIOW, Cdll StdLK, VdlldDie Vdlues	23 76	Z 76 s->leftover[s->leftover_offset++] = *(data++);									
70 conference of feet and a set of the set o	78	78 compress(s-viv, s->leftower); 79 s->leftower_offstat = 0;							uata now, call Stack, valiable values		
88 s=>bits_hashed += (TC_5HA256_BLOX_SIZE << 3);	88	80 s→bits_hushed ↔ (TC_5HA256_BLOCK_SIZE << 1); 31 3									



Mathematical Guarantees

TISA provides mathematically proven assurance of software safety, security, and reliability.



Efficient Compliance

TISA simplifies the compliance process with automated report generation, ensuring adherence to the most stringent industry standards.



Cost and Time Savings

TISA delivers comprehensive test coverage faster, reducing development time and costs.

TRUSTINSOFT ANALYZER DETECTS ALL UNDEFINED BEHAVIORS AND MORE

Buffer overflow
Use-after-free
Division by zero
Integer overflow
Array subscript out of range

•Strict aliasing violation •Dangerous function cast •Uninitialized memory •Memory leaks These bugs are subtle and complex to detect with typical testing methods. They are used for cyberattacks. They also introduce non-deterministic behaviors and cause software to crash and are used for cyber attacks

GUARANTEE ZERO-BUG SOFTWARE WITH TRUSTINSOFT ANALYZER

TrustInSoft is a leader in advanced software analysis tools and services that specializes in formal verification of C and C++ source code to ensure safety, security and reliability. Recognized by the US National Institute of Standards and Technology (NIST) for leveraging advanced formal methods, including abstract interpretation, TrustInSoft can mathematically guarantee analyzed software is free of critical runtime errors and vulnerabilities. TrustInSoft serves a diverse range of industries including automotive, aerospace, defense, consumer electronics, and IoT industries.



contact@trust-in-soft.com

EUROPE +33 1 84 06 43 91





trust-in-soft.com

