ID.	Described Floridades	Amabada	
ID	Result of Flawfinder	Analysis	
	Flawfinder version 2.0.17, (C) 2001-2019 David A. Wheeler. Number of rules (primarily dangerous function names) in C/C++ ruleset: 222		
	Examining src/videoStreamer.h		
	Examining src/videoStreamer.cpp		
	Examining src/common.cpp		
	Examining src/common.h		
	Examining src/NetworkTCP.h		
	Examining src/SecureCtx.cpp		
	Examining src/network.h		
	Examining src/pBox.h		
	Examining src/faceNet.h		
	Examining src/baseEngine.cpp		
	Examining src/SecureTCP.cpp		
	Examining src/TcpSendRecvJpeg.cpp		
	Examining src/faceNet.cpp		
	Examining src/baseEngine.h		
	Examining src/network.cpp		
	Examining src/onet_rt.h		
	Examining src/rnet_rt.h		
	Examining src/mtcnn.h		
	Examining src/TcpSendRecvJpeg.h		
	Examining src/onet_rt.cpp Examining src/pnet_rt.h		
	Examining src/rnet_rt.cpp		
	Examining src/mtcnn.cpp		
	Examining src/main.cpp		
	Examining src/pnet_rt.cpp		
	Examining src/NetworkTCP.cpp		
	Examining src/SecureCtx.h		
	Examining src/SecureTCP.h		
	FINAL RESULTS:		
	src/baseEngine.cpp:92: [2] (misc) open:		// Serialize engine
	Check when opening files - can an attacker redirect it (via symlinks),		ofstream planFile;
ID-1	force the opening of special file type (e.g., device files), move things		PlanFile.open(enginePath); //flawfinder_Sverflow : ignore - not considering an attacker redirect it— IHostMemory *serializedEngine = engine->serialize();
	around to create a race condition, control its ancestors, or change its	ignore-false pos	
	contents? (CWE-362).		
	src/common.h:68: [2] (buffer) char:	ignore-false	struct FaceRegion {
		positive	<pre>v int x1, x2, y1, y2; // left, right, top, bottom v bool isRegistered; v // wheter the detected face is for registered persion</pre>
ID-2	overflows or other issues (CWE-119!/CWE-120). Perform bounds checking,		» char userName[MAX_NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking
	functions that limit length, or ensure that the size is larger than the);
	maximum possible length.		
	src/faceNet.cpp:92: [2] (misc) open:	ignore-false	/* serialize engine and write to file */ if(m_serializeEngine) {
	Check when opening files - can an attacker redirect it (via symlinks),	positive	ofstream planFile;
ID-3	force the opening of special file type (e.g., device files), move things		planFile.open(m_engineFile); //flawfinder_5verflow : ignore - not considering an attacker redirect it
	around to create a race condition, control its ancestors, or change its		<pre>IHostMemory "serializedEngine = m_engine->serialize();</pre>
	contents? (CWE-362).		
	src/main.cpp:45: [2] (buffer) char:	ignore-false	char nameToRegister[NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking
	Chancelly Class arrays can be improperly received, localing to peteritar	positive	
ID-4	overflows or other issues (CWE-119!/CWE-120). Perform bounds checking,		
	functions that limit length, or ensure that the size is larger than the		
	maximum possible length.	Innere deles	<pre>void* socketChat(void *arg){</pre>
	src/main.cpp:328: [2] (buffer) char:	ignore-false positive	» // find if same connFd exist
ID-5	otatically-sized arrays can be improperly restricted, reading to potential	positive	» // if exist do send video
10-5	overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, functions that limit length, or ensure that the size is larger than the		unsigned char buffer[BUF_SIZE] = {0}; //flawfinder_Sverflow : ignore - perform bounds checking
	maximum possible length.		<pre>int retval;</pre>
	src/pnet_rt.cpp:24: [2] (misc) open:	ignore-false	void Pnet_engine::init(int row, int col) {
	Check when opening files - can an attacker redirect it (via symlinks),	positive	, , , , , , , , , , , , , , , , , , ,
ID-6	force the opening of special file type (e.g., device files), move things		<pre>//modifiy the input shape of prototxt, write to temp.prototxt int first_spce = 16, second_space = 4;</pre>
	around to create a race condition, control its ancestors, or change its		fstream protofile;
	contents? (CWE-362).		<pre>protofile.open(prototxt, ios::in); //flawfinder_Sverflow : ignore - not considering an attacker redirect it std::stringstream buffer;</pre>
	src/pnet_rt.cpp:41: [2] (misc) open:	ignore-false	protofile.close();
	Check when opening files - can an attacker redirect it (via symlinks),	positive	<pre>protofile.open("temp.prototxt", ios::out); //flawfinder_5verflow : ignore - not considering an attacker redirect it protofile.write(contents.c_str(), contents.size());</pre>
ID-7	force the opening of special file type (e.g., device files), move things		\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
	around to create a race condition, control its ancestors, or change its		
	contents? (CWE-362).		
	src/baseEngine.cpp:55: [1] (buffer) read:	ignore-false	std::cout << "size" << trtModelStreamsize() << std::endl; file_read(txtModelStreamdxtx() _size); //file_file_File_File_File_File_File_File_File_F
ID-8	Check buffer boundaries if used in a loop including recursive loops	positive	<pre>file.read(trtModelStreamdata(), size); //flawfinder_5verflow : ignore file.close();</pre>
	(CWE-120, CWE-20).		
	src/faceNet.cpp:43: [1] (buffer) read:	ignore-false	<pre>std::cout << "size" << 'trtModelStreamsize() << std::endl; file.read(trtModelStreamdata(), size); //flawfinder_5verflow : ignore</pre>
ID-9	Check buffer boundaries if used in a loop including recursive loops	positive	file.close();
	(CWE-120, CWE-20).	lanear f-t	unsigned char c;
ID-10	src/main.cpp:122: [1] (buffer) read:	ignore-false positive	<pre>if ((r = read(0, &c, sizeof(c))) < 0) { //flawfinder_5verflow : ignore</pre>
10-10	Check buffer boundaries if used in a loop including recursive loops (CWE-120, CWE-20).	,	return r;
	src/main.cpp:766: [1] (buffer) read:	ignore-false	
ID-11	Check buffer boundaries if used in a loop including recursive loops	positive	
.5 11	(CWE-120, CWE-20).		
	(22.1.2.1, 02.2).		
	ANALYSIS SUMMARY:		
	Hits = 11		
	Lines analyzed = 3456 in approximately 0.26 seconds (13211 lines/second)		
	Physical Source Lines of Code (SLOC) = 2577		
	Hits@level = [0] 57 [1] 4 [2] 7 [3] 0 [4] 0 [5] 0		
	Hits@level+ = [0+] 68 [1+] 11 [2+] 7 [3+] 0 [4+] 0 [5+] 0		
	Hits/KSLOC@level+ = [0+] 26.3873 [1+] 4.26853 [2+] 2.71634 [3+] 0 [4+]	0 [5+] 0	
	Minimum risk level = 1		
	Not every hit is necessarily a security vulnerability.		
	You can inhibit a report by adding a comment in this form:		
	// flawfinder: ignore		
	Make *sure* it's a false positive!		
	You can use the optionneverignore to show these.		
	There may be other security vulnerabilities; review your code!		
	See 'Secure Programming HOWTO'		

Nui Exa		Analysis	
Exa	wfinder version 2.0.17, (C) 2001-2019 David A. Wheeler.		
	mber of rules (primarily dangerous function names) in C/C++ rul		
Eve	amining src/videoStreamer.h		
⊏Xc	amining src/videoStreamer.cpp		
Exa	amining src/common.cpp		
Exa	amining src/common.h		
Exa	amining src/NetworkTCP.h		
	amining src/SecureCtx.cpp		
	amining src/network.h		
	amining src/pBox.h		
	amining src/faceNet.h		
	amining src/baseEngine.cpp		
Exa	amining src/SecureTCP.cpp		
Exa	amining src/TcpSendRecvJpeg.cpp		
Exa	amining src/faceNet.cpp		
Exa	amining src/baseEngine.h		
	amining src/network.cpp		
	amining src/onet_rt.h		
	amining src/rnet_rt.h		
	amining src/mtcnn.h		
	amining src/TcpSendRecvJpeg.h		
	amining src/onet_rt.cpp		
Exa	amining src/pnet_rt.h		
Exa	amining src/rnet_rt.cpp		
Exa	amining src/mtcnn.cpp		
Exa	amining src/main.cpp		
	amining src/pnet_rt.cpp		
	amining src/NetworkTCP.cpp		
	amining src/SecureCtx.h		
	-		
EX	amining src/SecureTCP.h		
	IAL DECLUTO.		
FIN	NAL RESULTS:		
			strlcpy(fr.userName,userName.c str(),sizeof(fr.userName)): // static analysis: strcnv to strlcnv
	:/faceNet.cpp:241: [4] (buffer) strcpy:	Modify to	strlcpy(fr.userName,userName.c_str(),sizeof(fr.userName)); // static analysis: strcpy to strlcpy
	es not check for buffer overflows when copying to destination [M	вігісру	* static analysis (flamfinder) * based on SEI CERT Coding Standard STB31-C. Guarantee that storage for strings has sufficient space for character data and the null terminator * buffer overflow is eliminated by removing strcpy() and calling the strlcpy(). * strlcpy is chosen for safe system since it guarantees Null Termination //
D-1			* Guarantee that storage for strings has sufficient space for character data and the null terminator * buffer overflow is eliminated by removing strong() and calling the strlong()
	NE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: s		* strlcpy is chosen for safe system since it guarantees Null Termination *
eas	sily misused).		//
	:/baseEngine.cpp:92: [2] (misc) open:	ignore-false	// Serialize engine
Che	eck when opening files - can an attacker redirect it (via symlinks	positive	ofstream planFile;
	ce the opening of special file type (e.g., device files), move thing		<pre>planFile.open(enginePath); //flawfinder_Sverflow : ignore - not considering an attacker redirect it— IHostNemory *serializedEngine = engine->serialize();</pre>
	ound to create a race condition, control its ancestors, or change i		
	ntents? (CWE-362).		
		ignore-false	struct FaceRegion {
	c/common.h:68: [2] (buffer) char:		» int x1, x2, y1, y2; // left, right, top, bottom
	atically-sized arrays can be improperly restricted, leading to pote		» bool isRegistered;» // wheter the detected face is for registered persion » char userName[NAX_NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking
	erflows or other issues (CWE-119!/CWE-120). Perform bounds of))
fun	ctions that limit length, or ensure that the size is larger than the		
ma	ximum possible length.		
src	:/faceNet.cpp:92: [2] (misc) open:	ignore-false	/* serialize engine and write to file */
	eck when opening files - can an attacker redirect it (via symlinks		<pre>if(m_serializeEngine) { ofstream planFile;</pre>
	ce the opening of special file type (e.g., device files), move thing		planFile.open(m_engineFile); //flawfinder_5verflow : ignore - not considering an attacker redirect it
	ound to create a race condition, control its ancestors, or change i		<pre>IHostMemory *serializedEngine = m_engine->serialize();</pre>
	ntents? (CWE-362).		
		Madifica	<pre>snprintf(fr.userName, sizeof(fr.userName), "Unknown"); //default // static analysis: sprintf to snprintf</pre>
	c/faceNet.cpp:233: [2] (buffer) sprintf:	Modify to	
	es not check for buffer overflows (CWE-120). Use sprintf_s, snp	onprint.	* static analysis (flawfinder) * based on SEI CERT C Coding Standard STR31-C.
Do			
			 Guarantee that storage for strings has sufficient space for character data and the null terminator
Do			* Guarantee that storage for strings has sufficient space for character data and the null terminator * buffer overflow is eliminated by removing sprintf() and calling the snprintf()
Doo	printf. Risk is low because the source has a constant maximum		Guarantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf()
Doo D-5	nprintf. Risk is low because the source has a constant maximum		Guarantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf()
D-5 vsn	/main.cpp:45: [2] (buffer) char:	ignore-false	"Guarantee that storage for strings has sufficient space for character data and the null terminator "buffer overflow is eliminated by removing sprintf() and calling the snprintf() char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking
Door Vsn src Sta	./main.cpp:45: [2] (buffer) char: stically-sized arrays can be improperly restricted, leading to pote	ignore-false positive	Guarantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf()
Door vsn src Sta	Imain.cpp:45: [2] (buffer) char: tically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds of	ignore-false positive	Guarantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf()
Vsn vsn src Sta D-6 ove	Imain.cpp:45: [2] (buffer) char: tically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the	ignore-false positive	Guarantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf()
Door vsn src Sta D-6 over fun ma	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds of citions that limit length, or ensure that the size is larger than the ximum possible length.	ignore-false positive	* Guarantee that storage for strings has sufficient space for character data and the null terminator *buffer overflow is eliminated by removing sprintf() and calling the snprintf() / char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking [void* SOCKetChat(void *arg){
Door vsn src Sta D-6 over fun ma	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds c ctions that limit length, or ensure that the size is larger than the ximum possible length.	ignore-false positive ignore-false	**Guarantee that storage for strings has sufficient space for character data and the null terminator **buffer overflow is eliminated by removing sprintf() and calling the snprintf() // **Char nameToRegister[NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking **Jevoid** SocketChat(void *arg){ **But is ame connfd exist.**
vsn src Sta D-6 ove fun ma src Sta	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote striflows or other issues (CWE-119/CWE-120). Perform bounds cotions that limit length, or ensure that the size is larger than the ximum possible length. Imain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote	ignore-false positive ignore-false positive	* Guarantee that storage for strings has sufficient space for character data and the null terminator *buffer overflow is eliminated by removing sprintf() and calling the snprintf() / char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking [void* SOCKetChat(void *arg){
VSn Sta D-6 Over Sta	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: tically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o	ignore-false positive ignore-false positive	"Superantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf() char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* SOCKEtChat(void *arg){ // find if same connfd exist // if exist do send video unsigned char buffer(BUF_SIZE) = {0}; //flawfinder_Sverflow : ignore - perform bounds checking
VSn Sta D-6 Over Sta	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote striflows or other issues (CWE-119/CWE-120). Perform bounds cotions that limit length, or ensure that the size is larger than the ximum possible length. Imain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote	ignore-false positive ignore-false positive	Guarantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf() char nameToRegister[NAME_LEN]; //flawfinder_Sverflow: ignore - perform bounds checking livoid* SocketChat(void *arg){ // find if same connfd exist // if exist do send video
vsn src Sta D-6 ove fun ma src Sta D-7 ove fun	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: tically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o	ignore-false positive ignore-false positive	"Superantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf() char nameToRegister[NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking livoid * SocketChat(void *arg){ // find if same connfd exist // if exist do send video unsigned char buffer[BUF_SIZE] = {0}; //flawfinder_5verflow : ignore - perform bounds checking int retval;
vsn src Sta D-6 ove fun ma src Sta D-7 fun ma	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: tically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. //mtcnn.cpp:125: [2] (buffer) memcpy:	ignore-false positive ignore-false positive	"Superantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf() char nameToRegister[NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking livoid* SOCKEtChat(void *arg){ // find if same connfd exist // if exist do send video unsigned char buffer[BUF_SIZE] = {0}; //flawfinder_5verflow : ignore - perform bounds checking if(*(refineNet->score>pdata*)>refineNet->Rthreshold){
vsn src Sta D-6 ove fun ma src Sta D-7 ove fun ma	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: tically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. //mtcnn.cpp:125: [2] (buffer) memcpy:	ignore-false positive ignore-false positive Modify destination	"Superantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf() char nameToRegister[NWE_LEN]; //flawfinder_5verflow : ignore - perform bounds checking livoid* SOCKEtChat(void *arg){ // find if same connfd exist // find if same connfd exist // if exist do send videe unsigned char buffer[BUF_SIZE] = {0}; //flawfinder_5verflow : ignore - perform bounds checking int retval; if(*(refineNet->score)pdata*1))refineNet->Rthreshold){ ///samcpy(it->regrecord, refineNet->location>pdata, 4*sizeof(mydata#nt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120)
vsn src Sta D-6 ove fun ma src Sta D-7 ove fun ma	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. Ermain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length.	ignore-false positive ignore-false positive	**Guarantee that storage for strings has sufficient space for character data and the null terminator **buffer overflow is eliminated by removing sprintf() and calling the snprintf() char nameToRegister[NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking livoid **socketChat(void *arg){
vsn src Sta D-6 ove fun ma src Sta D-7 ove fun fun Sta D-7 ove fun fun Src D-8	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: tically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. //mtcnn.cpp:125: [2] (buffer) memcpy:	ignore-false positive ignore-false positive Modify destination	"Superantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf() char nameToRegister[NWE_LEN]; //flawfinder_5verflow : ignore - perform bounds checking livoid* SOCKEtChat(void *arg){ // find if same connfd exist // find if same connfd exist // if exist do send videe unsigned char buffer[BUF_SIZE] = {0}; //flawfinder_5verflow : ignore - perform bounds checking int retval; if(*(refineNet->score)pdata*1))refineNet->Rthreshold){ ///samcpy(it->regrecord, refineNet->location>pdata, 4*sizeof(mydata#nt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120)
vsn src src Sta D-6 over fun ma src Sta D-7 over fun ma src Sta D-7 Ma Ma	Emain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. Zimain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. Zimtcnn.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data.	ignore-false positive ignore-false positive Modify destination length	"Superantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliainated by removing sprintf() and calling the snprintf() char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* SocketChat(void *arg){
Dool vsn src Sta Dool Over Sta	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds of ctions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds of ctions that limit length, or ensure that the size is larger than the ximum possible length. //mtcnn.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtcnn.cpp:157: [2] (buffer) memcpy:	ignore-false positive ignore-false positive Modify destination length Modify	"Suprantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf() char nameToRegister[NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking livoid* socketChat(void *arg){ // fland if same connfd exist // ffare visit do send video unsigned char buffer[BUF_SIZE] = {8}; //flawfinder_5verflow : ignore - perform bounds checking if(*(refineNet->score>pdata*1)>refineNet->Rthreshold){ //semcpy(it->regreCoord, refineNet->location>pdata, 4*sizeof(mydata*mt)); //flawfinder_5verflow : boes not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regreCoord, refineNet->location>pdata, 4*sizeof(it->regreCoord)); if(*(outNet->score>pdata*1)>outNet->Othreshold){ //memcpy(it->regreCoord, outNet->location>pdata, 4*sizeof(mydata*mt)); if(*(outNet->score>pdata*1)>outNet->Othreshold){ //memcpy(it->regreCoord, outNet->location>pdata, 4*sizeof(mydata*mt));
Dool vsn src Sta D-6 over fun ma src Sta D-7 over fun ma src Sta D-7 over fun ma src Ma src	Emain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. Zimain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. Zimtcnn.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data.	ignore-false positive ignore-false positive Modify destination length Modify	"Superantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliainated by removing sprintf() and calling the snprintf() char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* SocketChat(void *arg){
Do-5 vsn src Sta D-6 over fun ma src Sta D-7 over fun pola src Do-8 Ma src Do-9	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtcnn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C	ignore-false positive ignore-false positive Modify destination length Modify destination	char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* SocketChat(void *arg){ // find if same connfd exist // if exist do send video unsigned char buffer[BUF_SIZE] = {0}; //flawfinder_Sverflow : ignore - perform bounds checking if(*(refineNet->score>pdata*i)>refineNet->Nethershold){ //esency(it->regrecoord, refineNet->Nethershold){ //esency(it->regrecoord, refineNet->Contion>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) if(*(votNet->score>pdata*i)>outNet->Oothreshold){ //mency(it->regrecoord, outNet->Oothreshold){ //mency(it->regrecoord, outNet->Oothreshold){ //mency(it->regrecoord, outNet->Oothreshold){ //mency(it->regrecoord, outNet->Oothreshold){ //mency(it->regrecoord, outNet->Oothreshold){ //mency(it->regrecoord, outNet->Oothreshold){ //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120)
Do-5 vsn src Sta D-6 ove fun ma src Sta D-7 fun ma src Don Ma src Do-9 Ma	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds o cotions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:328: [2] (buffer) char: tically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds o cotions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data.	ignore-false positive ignore-false positive Modify destination length Modify destination length	"Suprantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf() char nameToRegister(NAME_LEN); //flawfinder_Sverflow : ignore - perform bounds checking lvoid* socketChat(void *arg){ // find if same connfd exist // if exist do send video unsigned char buffer(BUF_SIZE) = {0}; //flawfinder_Sverflow : ignore - perform bounds checking int retval; if(*(refineNet->score>pdata=i)>refineNet->Rthreshold){ //seecpy(it->regreCoord, refineNet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regreCoord, outlet->location>pdata, 4*sizeof(mydataFmt)); //seecpy(it->regreCoord, outlet->location>pdata, 4*sizeof(it->regreCoord));
Do-5 vsn src Sta D-6 oven fun ma src Sta D-7 oven fun ma src Sta D-7 oven fun ma src D-7 fun ma src Do-9 Ma src	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds of citions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds of citions that limit length, or ensure that the size is larger than the ximum possible length. //mtcnn.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtcnn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtcnn.cpp:157: [2] (buffer) memcpy:	ignore-false positive ignore-false positive Modify destination length Modify destination length ignore-false	"Suprantee that storage for strings has sufficient space for character data and the null terminator buffer overflow is eliminated by removing sprintf() and calling the snprintf() char nameToRegister[NaVE_LEN]; //flawfinder_5verflow : ignore - perform bounds checking lvoid* socketChat(void *arg){ // find if same connfd exist // find if same connfd exist // if exist do send video unsigned char buffer[BUF_SIZE] = {8}; //flawfinder_5verflow : ignore - perform bounds checking int retval; if(*(refineNet->score>pdata*1)>refineNet->Rthreshold){ //semcpy(it->regreCoord, refineNet->location>pdata, 4*sizeof(mydataFatt)); //flawfinder_5verflow : Does not theck for buffer overflows when copying to destination (CNE-120) memcpy(it->regreCoord, outlet->location>pdata, 4*sizeof(mydataFatt)); //flawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regreCoord, outlet->location>pdata, 4*sizeof(mydataFatt)); //flawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regreCoord, outlet->location>pdata, 4*sizeof(it->regreCoord)); void Pnet_engine::init(int row, int col) {
Do-5 vsn src Sta D-6 ove fun ma src Sta D-7 ove fun ma src D-7 ove fun ma src D-7 ove fun ma src Chc Chc	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. //mtcnn.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtcnn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //pnet_rt.cpp:24: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks	ignore-false positive ignore-false positive Modify destination length Modify destination length ignore-false positive	char nameToRegister[NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking char nameToRegister[NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking void* socketChat(void *arg){ // find if same counfd exist // if exist do send video unsigned char buffer[BUF_SIZE] = {0}; //flawfinder_5verflow : ignore - perform bounds checking iif (*(refinellet->score_>pdata*i)>refinellet->Rthreshold){ //memcpy(it->regrectoord, refinellet->location_>pdata, 4*sizeof(mydata*mt)); //flawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrectoord, outhlet->location_>pdata, 4*sizeof(mydata*mt)); //flawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrectoord, outhlet->location_>pdata, 4*sizeof(mydata*mt)); //flawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrectoord, outhlet->location_>pdata, 4*sizeof(mydata*mt)); //filawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrectoord, outhlet->location_>pdata, 4*sizeof(it->regrectoord)); void Pnet_engine::init(int row, int col) { //maddfiy the input shape of prototxt, write to temp.prototxt
Doi-5 vsn	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnp:157: [2] (misc) open: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnp:157: [2] (misc) open: es not check for buffer overflows when copying to destination (C	ignore-false positive ignore-false positive Modify destination length Modify destination length ignore-false positive	char nameToRegister[NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking lvoid* socketChat(void *arg){ // find if same comrid exist // if exist do send video unsigned char buffer[BUF_SIZE] = {0}; //flawfinder_5verflow : ignore - perform bounds checking iif(*(refinelet->score>pdata*))refinelet->Rthreshold){ //senergy(it->regrecoord, refinelet->Rthreshold){ //senergy(it->regrecoord, refinelet->location>pdata, 4*sizeof(it->regrecoord)); if(*(outNet->score>pdata*))>outNet->Othershold){ //senergy(it->regrecoord, refinelet->location>pdata, 4*sizeof(it->regrecoord)); if(*(outNet->score>pdata*))>outNet->Othershold){ //senergy(it->regrecoord, outlet->location>pdata, 4*sizeof(it->regrecoord)); //flawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outlet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outlet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outlet->location>pdata, 4*sizeof(it->regrecoord)); void Pnet_engine::init(int_row, int col) { //sendify the input shape of prototxt, write to temp.prototxt int first_spce_ = 16, second_space_ = 4; fstream protofile;
vsn	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. //mtcnn.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtcnn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //pnet_rt.cpp:24: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks	ignore-false positive ignore-false positive Modify destination length Modify destination length ignore-false positive	char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* socketChat(void *arg){ // find if same countd exist // if exist do send video unsigned char buffer[BUF_SIZE] = {0}; //flawfinder_Sverflow : ignore - perform bounds checking if(*(refineNet->score>pdata*i)>refineNet->Rthreshold){ //sencpy(it->regrecoord, refineNet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) mescpy(it->regrecoord, outlet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) mescpy(it->regrecoord, outlet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) mencpy(it->regrecoord, outlet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) mencpy(it->regrecoord, outlet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) mencpy(it->regrecoord, outlet->location>pdata, 4*sizeof(it->regrecoord)); void Pnet_engine::init(int row, int col) { //modify the input shape of prototxt, write to temp.prototxt irt first spec_n=16, second_space = 4; protofile.open(prototxt, iss::in): //flawfinder Sverflow : isnore - not considering an attacker redirect it
vsn	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cpp:14: [2] (misc) open: es when opening files - can an attacker redirect it (via symlinks ce the opening of special file type (e.g., device files), move thing und to create a race condition, control its ancestors, or change i	ignore-false positive ignore-false positive Modify destination length Modify destination length ignore-false positive	char nameToRegister[NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking lvoid* socketChat(void *arg){ // find if same comrid exist // if exist do send video unsigned char buffer[BUF_SIZE] = {0}; //flawfinder_5verflow : ignore - perform bounds checking iif(*(refinelet->score>pdata*))refinelet->Rthreshold){ //senergy(it->regrecoord, refinelet->Rthreshold){ //senergy(it->regrecoord, refinelet->location>pdata, 4*sizeof(it->regrecoord)); if(*(outNet->score>pdata*))>outNet->Othershold){ //senergy(it->regrecoord, refinelet->location>pdata, 4*sizeof(it->regrecoord)); if(*(outNet->score>pdata*))>outNet->Othershold){ //senergy(it->regrecoord, outlet->location>pdata, 4*sizeof(it->regrecoord)); //flawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outlet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outlet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outlet->location>pdata, 4*sizeof(it->regrecoord)); void Pnet_engine::init(int_row, int col) { //sendify the input shape of prototxt, write to temp.prototxt int first_spce_ = 16, second_space_ = 4; fstream protofile;
vsn	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o cotions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o cotions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cpp:14: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks ce the opening of special file type (e.g., device files), move thing and to create a race condition, control its ancestors, or change i nenters? (CWE-362).	ignore-false positive ignore-false positive Modify destination length Modify destination length ignore-false positive	"Suprementation is climinated by removing sprint(f) and calling the snprint(f) buffer overflow is climinated by removing sprint(f) and calling the snprint(f) can call and the null terminator buffer sprint(f) and calling the snprint(f) can call and the suprime sprint(f) can call and the suprime sprint(f) can call an experiment call an
Doi-5	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds of ctions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds of ctions that limit length, or ensure that the size is larger than the ximum possible length. //mtenn.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtenn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //protent-r.cpp:24: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing und to create a race condition, control its ancestors, or change i tentents? (CWE-362). //pnet_rt.cpp:41: [2] (misc) open:	ignore-false positive ignore-false positive Modify destination length Modify destination length ignore-false positive	char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* SocketChat(void *arg){ // find if same connfd exist // if exist do send video // if exist do s
Dot	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds of ctions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds of ctions that limit length, or ensure that the size is larger than the ximum possible length. //mtcnn.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtcnn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //pnet_rt.cpp:24: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing pund to create a race condition, control its ancestors, or change i ntents? (CWE-362). //pnet_rt.cpp:41: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks) eck when opening files - can an attacker redirect it (via symlinks) extended the sure and the sure of t	ignore-false positive ignore-false positive Modify destination length Modify destination length ignore-false positive	char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* socketChat(void *arg){
Dot Dot	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds of ctions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds of ctions that limit length, or ensure that the size is larger than the ximum possible length. //mtenn.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtenn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //protent-r.cpp:24: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing und to create a race condition, control its ancestors, or change i tentents? (CWE-362). //pnet_rt.cpp:41: [2] (misc) open:	ignore-false positive ignore-false positive Modify destination length Modify destination length ignore-false positive	char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* SocketChat(void *arg){ // find if same connfd exist // if exist do send video // if exist do s
Dot Dot	Imain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds of ctions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119!/CWE-120). Perform bounds of ctions that limit length, or ensure that the size is larger than the ximum possible length. //mtcnn.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtcnn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //pnet_rt.cpp:24: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing pund to create a race condition, control its ancestors, or change i ntents? (CWE-362). //pnet_rt.cpp:41: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks contents? (CWE-362).	ignore-false positive ignore-false positive Modify destination length Modify destination length ignore-false positive	char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* SocketChat(void *arg){ // find if same connfd exist // if exist do send video // if exist do s
Dot Dot	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cpp:14: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing and to create a race condition, control its ancestors, or change i altents? (CWE-362). chain.cpp:41: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing and to general of special file type (e.g., device files), move thing and to create a race condition, control its ancestors, or change i and to create a race condition, control its ancestors, or change i and to create a race condition, control its ancestors, or change i and to create a race condition, control its ancestors, or change i	ignore-false positive ignore-false positive Modify destination length Modify destination length ignore-false positive	char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* SocketChat(void *arg){ // find if same connfd exist // if exist do send video // if exist do s
Dot Dot	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds of ctions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds of ctions that limit length, or ensure that the size is larger than the ximum possible length. //mtenn.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtenn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtenn.cpp:24: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks tee the opening of special file type (e.g., device files), move thing und to create a race condition, control its ancestors, or change i thents? (CWE-362). //pnet_rt.cpp:41: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing und to create a race condition, control its ancestors, or change i ntents? (CWE-362).	ignore-false positive Modify destination length Modify destination length ignore-false positive	char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* SocketChat(void *arg){ // find if same connfd exist // if exist do send video unsigned char buffer(BUF_SIZE) = (0); //flawfinder_Sverflow : ignore - perform bounds checking if(*(refineNet->score>pdatail)>refineNet->Rhreshold){ //esencpy(it->regrecoord, refineNet->Duffershold); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) mencpy(it->regrecoord, refineNet->Duffershold){ //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) mencpy(it->regrecoord, outlet->Outhreshold){ //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) mencpy(it->regrecoord, outlet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) mencpy(it->regrecoord, outlet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) mencpy(it->regrecoord, outlet->location>pdata, 4*sizeof(it->regrecoord)); void Pnet_engine::init(int row, int col) { //modify the input shape of prototxt, write to temp.prototxt int first space = 16, second space = 4; fstream protofile. protofile.close(); protofile.close(); protofile.close(); protofile.close(); protofile.open('temp.prototxt', ios::out); //flawfinder_Sverflow : ignore - not considering an attacker redirect it protofile.write(contents.c_str(), contents.size());
Dot State	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnp:157: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks the the opening of special file type (e.g., device files), move thing and to create a race condition, control its ancestors, or change in thents? (CWE-362). chain.cpp:41: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing and to create a race condition, control its ancestors, or change in thents? (CWE-362). change [CWE-362].	ignore-false positive Ignore-false positive Modify destination length Modify destination length ignore-false positive	char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* socketChat(void *arg){ // find if same connfd exist // if exist do send video // unsigned char buffer[SUF_SIZE] = {0}; //flawfinder_Sverflow : ignore - perform bounds checking if(*(refineNet->score>pdata*i)>refineNet->Nthreshold){ //esecpy(it->regrecoord, refineNet->Dotthreshold){ //esecpy(it->regrecoord, refineNet->Dotthreshold){ //memcpy(it->regrecoord, refineNet->Dotthreshold){ //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) //flawfinder_Sverflow : ignore - not considering an attacker redirect it std::ettingstream buffer; protofile.close(); //flawfinder_Sverflow : ignore - not considering an attacker redirect it std::ettingstream buffer; protofile.copen('temp_prototxt', ios::out); //flawfinder_Sverflow : ignore - not considering an attacker redirect it protofile.wite(contents.c_str(), contents.size() < std::endl; file.read(trithodelStreamsize() < std::endl; file.read(trithodelStreamdata(), size); //flawfinder_Sverflow : ignore
Dot Single Dot	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o cotions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o cotions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chance, pr. 157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chance, pr. 157: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks coe the opening of special file type (e.g., device files), move thing and to create a race condition, control its ancestors, or change in thents? (CWE-362). chance, pr. 41: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks coe the opening of special file type (e.g., device files), move thing and to create a race condition, control its ancestors, or change in thents? (CWE-362). chance files, move thing and to create a race condition, control its ancestors, or change in thents? (CWE-362). chance files, move thing and to create a race condition, control its ancestors, or change in thents? (CWE-362).	ignore-false positive Ignore-false positive Modify destination length Modify destination length ignore-false positive	char nameToRegister(NAME_LEN); //flawfinder_5verflow : ignore - perform bounds checking char nameToRegister(NAME_LEN); //flawfinder_5verflow : ignore - perform bounds checking lvoid* socketChat(void *arg){ // find if same counfd exist // if if wist do send video unsigned char buffer(BUF_SIZE] = {0}; //flawfinder_5verflow : ignore - perform bounds checking iif(*(refinelet->score>pdata*))*refinelet->Rthreshold){ //senepy(it->regrecoord, refinelet->Rthreshold){ //senepy(it->regrecoord, refinelet->Rthreshold){ //senepy(it->regrecoord, refinelet->location>pdata, 4*sizeof(it->regrecoord)); if(*(outNet->score>pdata*))>outNet->Othreshold){ //mencpy(it->regrecoord, outNet->location>pdata, 4*sizeof(it->regrecoord)); if(*(outNet-)*score>pdata*))>outNet->Othreshold){ //mencpy(it->regrecoord, outNet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) mencpy(it->regrecoord, outNet->location>pdata, 4*sizeof(mydataFmt)); //flawfinder_5verflow : Does not check for buffer overflows when copying to destination (CNE-120) mencpy(it->regrecoord, outNet->location>pdata, 4*sizeof(it->regrecoord)); void Pnet_engine::init(int row, int col) { //modifyly the input shape of prototxt, write to temp.prototxt int first_spce = 16, second_space = 4; fstream_protofile; protofile.open(prototxt, ios::in); //flawfinder_5verflow : ignore - not considering an attacker redirect it std::cout << *size* << trth>trtModelStreamsize() << std::endl; std::cout << *size* << trth>trtModelStreamsize() << std::endl;
Doc State	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. //mtenn.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtenn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtenn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //prot_rt.cpp:24: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing fund to create a race condition, control its ancestors, or change i ntents? (CWE-362). //prot_rt.cpp:41: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing nund to create a race condition, control its ancestors, or change i ntents? (CWE-362). //baseEngine.cpp:55: [1] (buffer) read: eck buffer boundaries if used in a loop including recursive loops //E-120, CWE-20).	ignore-false positive Modify destination length Modify destination length ignore-false positive ignore-false positive	char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* SocketChat(void *arg){ // find if same connfd exist // if exist do send video // unsigned char buffer[SUF_SIZE] = {0}; //flawfinder_Sverflow : ignore - perform bounds checking if(*(refineNet->score>pdatail)*refineNet->Rhreshold){ ///seencpy(it->regrecoord, refineNet->Dethreshold){ ///seencpy(it->regrecoord, refineNet->Dethreshold){ ///seencpy(it->regrecoord, outNet->Oothreshold){ ///seencpy(it->regrecoord, outNet->Jocation>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outNet->Jocation>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outNet->Jocation>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outNet->Jocation>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outNet->Jocation>pdata, 4*sizeof(it->regrecoord)); void Pnet_engine::init(int row, int col) { //modify the input shape of prototxt, write to temp.prototxt int first spec = 16, second space = 4; fitream protofile; protofile.close(); protofile.close(); protofile.close(); protofile.write(contents.c_str(), contents.size()); std::cout < <a "="" href="">**size** < < *trtModelStreamsize() *std::cout *size** < *trtModelStreamsize() *std::cout *std::cout *size** < *trtModelStreamsize() *std::cout *std::cout *size** < *trtModelStreamsize() *std::cout *std::cout *size** < *trtModelStreamsize() <a ""="" href="</td></tr><tr><th> Dot Dot </th><td>Emain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. Imain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. Imain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. Immon.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. Immon.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. Immon.cpp:157: [2] (misc) open: es when opening files - can an attacker redirect it (via symlinks the the opening of special file type (e.g., device files), move thing sund to create a race condition, control its ancestors, or change in thents? (CWE-362). Intents? (CWE-362).</td><td>ignore-false positive Modify destination length Ignore-false positive Modify destination length Ignore-false positive Ignore-false positive Ignore-false positive</td><td>char nameToRegister[NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking char nameToRegister[NAME_LEN]; //flawfinder_5verflow : ignore - perform bounds checking lvoid* socketChat(void *arg){</td></tr><tr><th> Doi: Doi: </th><td>chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. //main.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o citions that limit length, or ensure that the size is larger than the ximum possible length. //mtenn.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtenn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //mtenn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. //prot_rt.cpp:24: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing fund to create a race condition, control its ancestors, or change i ntents? (CWE-362). //prot_rt.cpp:41: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing nund to create a race condition, control its ancestors, or change i ntents? (CWE-362). //baseEngine.cpp:55: [1] (buffer) read: eck buffer boundaries if used in a loop including recursive loops //E-120, CWE-20).</td><td>ignore-false positive Modify destination length Ignore-false positive Modify destination length Ignore-false positive Ignore-false positive Ignore-false positive</td><td>char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* SocketChat(void *arg){ // find if same connfd exist // if exist do send video // unsigned char buffer[SUF_SIZE] = {0}; //flawfinder_Sverflow : ignore - perform bounds checking if(*(refineNet->score>pdatail)*refineNet->Rhreshold){ ///seencpy(it->regrecoord, refineNet->Dethreshold){ ///seencpy(it->regrecoord, refineNet->Dethreshold){ ///seencpy(it->regrecoord, outNet->Oothreshold){ ///seencpy(it->regrecoord, outNet->Jocation>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outNet->Jocation>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outNet->Jocation>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outNet->Jocation>pdata, 4*sizeof(mydataFmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regrecoord, outNet->Jocation>pdata, 4*sizeof(it->regrecoord)); void Pnet_engine::init(int row, int col) { //modify the input shape of prototxt, write to temp.prototxt int first spec = 16, second space = 4; fitream protofile; protofile.close(); protofile.close(); protofile.close(); protofile.write(contents.c_str(), contents.size()); std::cout <**size** < < *trtModelStreamsize() *std::cout *size** < *trtModelStreamsize() *std::cout *std::cout *size** < *trtModelStreamsize() *std::cout *std::cout *size** < *trtModelStreamsize() *std::cout *std::cout *size** < *trtModelStreamsize() <a :="" <<="" c;="" c;<="" char="" file.close();="" flawfinder_sverflow="" href="</td></tr><tr><th> Doi-5 Doi-5 </th><td>Emain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. Imain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. Imain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. Immon.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. Immon.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. Immon.cpp:157: [2] (misc) open: es when opening files - can an attacker redirect it (via symlinks the the opening of special file type (e.g., device files), move thing sund to create a race condition, control its ancestors, or change in thents? (CWE-362). Intents? (CWE-362).</td><td>ignore-false positive Modify destination length Ignore-false positive Modify destination length Ignore-false positive Ignore-false positive Ignore-false positive</td><td>char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking livoid* socketChat(void *arg){ // find if same connfd exist // if exist do send video // if exist do s</td></tr><tr><th> Doi: Doi: </th><td>chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o cotions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o cotions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chance, check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chance, check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chance theological file to can an an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing pund to create a race condition, control its ancestors, or change in tentents? (CWE-362). chaseEngine.cpp:355: [1] (buffer) read: eck buffer boundaries if used in a loop including recursive loops WE-120, CWE-20). chaseIngine.cpp:43: [1] (buffer) read: eck buffer boundaries if used in a loop including recursive loops</td><td>ignore-false positive Modify destination length Ignore-false positive Modify destination length Ignore-false positive Ignore-false positive Ignore-false positive</td><td>char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking // char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking // char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking // find if same connfd exist // if exist do send video unsigned char buffer[BUF_SIZE] = {0}; //flawfinder_Sverflow : ignore - perform bounds checking if(*(refinelet->score>pdatail)>refinelet->Rthreshold){ //memcpy(it->regreCoord, refinelet->location>pdata, 4*sizeof(aydatafmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regreCoord, orefinelet->location>pdata, 4*sizeof(mydatafmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regreCoord, outlet->Othreshold){ //memcpy(it->regreCoord, outlet->Othreshold){ //memcpy(it->regreCoord, outlet->Othreshold){ //memcpy(it->regreCoord, outlet->Othreshold){ //memcpy(it->regreCoord, outlet->Othreshold){ //memcpy(it->regreCoord, outlet->Othreshold){ //memcpy(it->regreCoord, outlet->Decation>pdata, 4*sizeof(mydatafmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regreCoord, outlet->Decation>pdata, 4*sizeof(mydatafmt)); //flawfinder_Sverflow : ignore - not considering an attacker redirect it int first spce = 16, second_space = 4; fstream protofile; portofile.close(); protofile.close(); protofile.close(); protofile.copen(prototxt, ios::on); //flawfinder_Sverflow : ignore - not considering an attacker redirect it protofile.write(contents.c_str(), contents.size() < std::endl; file.close(); std::cout << " ignore="" size"="" size);="" td="" trtmodelstreamdata(),="" unsigned="">
Dot Dot	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cpp:157: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing tund to create a race condition, control its ancestors, or change i ntents? (CWE-362). chaels? (CWE-362). chaels? (CWE-362). chaels are condition, control its ancestors, or change i ntents? (CWE-362). chaels are condition, control its ancestors, or change i ntents? (CWE-362). chaels are condition, control its ancestors, or change i ntents? (CWE-362). chaels are condition, control its ancestors, or change i ntents? (CWE-362). chaels are condition, control its ancestors, or change i ntents? (CWE-362). chaels are condition, control its ancestors, or change i ntents? (CWE-362). chaels are condition, control its ancestors, or change i ntents? (CWE-362). chaels are condition, control its ancestors, or change i ntents? (CWE-362). chaels are condition, control its ancestors, or change i ntents? (CWE-362). chaels are chaels are condition, control its ancestors, or change i ntents? (C	ignore-false positive Modify destination length destination length ignore-false positive ignore-false positive ignore-false positive ignore-false positive ignore-false positive	Char nameToRegister[NWME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking char nameToRegister[NWME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking // find if same conned exist // if exist do send video // int retval; // if exist do send video // int retval; // if exist do send video // int retval; // // if exist do send video // int retval; // // if exist do send video // int retval; // // if exist do send video // int retval; // // if exist do send video // int retval; // // if exist do send video // int retval; // in
Dot Dot	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. Imain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. Imain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o ctions that limit length, or ensure that the size is larger than the ximum possible length. Internacion composition of the size is larger than the ximum possible length. Internacion composition can always hold the source data. Internacion composition can always h	ignore-false positive Modify destination length destination length ignore-false positive ignore-false positive ignore-false positive ignore-false positive ignore-false positive	char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking // char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking // char nameToRegister[NAME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking // find if same connfd exist // if exist do send video unsigned char buffer[BUF_SIZE] = {0}; //flawfinder_Sverflow : ignore - perform bounds checking if(*(refinelet->score>pdatail)>refinelet->Rthreshold){ //memcpy(it->regreCoord, refinelet->location>pdata, 4*sizeof(aydatafmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regreCoord, orefinelet->location>pdata, 4*sizeof(mydatafmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regreCoord, outlet->Othreshold){ //memcpy(it->regreCoord, outlet->Othreshold){ //memcpy(it->regreCoord, outlet->Othreshold){ //memcpy(it->regreCoord, outlet->Othreshold){ //memcpy(it->regreCoord, outlet->Othreshold){ //memcpy(it->regreCoord, outlet->Othreshold){ //memcpy(it->regreCoord, outlet->Decation>pdata, 4*sizeof(mydatafmt)); //flawfinder_Sverflow : Does not check for buffer overflows when copying to destination (CNE-120) memcpy(it->regreCoord, outlet->Decation>pdata, 4*sizeof(mydatafmt)); //flawfinder_Sverflow : ignore - not considering an attacker redirect it int first spce = 16, second_space = 4; fstream protofile; portofile.close(); protofile.close(); protofile.close(); protofile.copen(prototxt, ios::on); //flawfinder_Sverflow : ignore - not considering an attacker redirect it protofile.write(contents.c_str(), contents.size() < std::endl; file.close(); std::cout << "size" << trtModelStreamdata(), size); //flawfinder_Sverflow : ignore file.close(); unsigned char c; unsigned char c; unsigned char c;
Dot State Dot	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o cotions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o cotions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnp:125: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks coe the opening of special file type (e.g., device files), move thing and to create a race condition, control its ancestors, or change i ments? (CWE-362). chain.cp:121: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks coe the opening of special file type (e.g., device files), move thing and to create a race condition, control its ancestors, or change i ments? (CWE-362). chain.cp:125: [1] (buffer) read: eck buffer boundaries if used in a loop including recursive loops WE-120, CWE-20). chain.cpp:122: [1] (buffer) read: eck buffer boundaries if used in a loop including recursive loops WE-120, CWE-20).	ignore-false positive Modify destination length Modify destination length ignore-false positive ignore-false positive ignore-false positive ignore-false positive ignore-false positive ignore-false positive	Char nameToRegister[NUME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking char nameToRegister[NUME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking lovoid* socketChat(void *arg){
Dot Dot	Emain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds of citions that limit length, or ensure that the size is larger than the ximum possible length. Imain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds of citions that limit length, or ensure that the size is larger than the ximum possible length. Imain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. Immtenn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. Immtenn.cpp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. Immtenn.cpp:147: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks to the opening of special file type (e.g., device files), move thing that oreate a race condition, control its ancestors, or change in thents? (CWE-362). Immtense (CWE-20). Immain.cpp:122: [1] (buffer) read: eck buffer boundaries if used in a loop including recursive loops WE-120, CWE-20). Immain.cpp:377: [1] (buffer) strncpy:	ignore-false positive Modify destination length Modify destination length ignore-false positive ignore-false positive ignore-false positive ignore-false positive ignore-false positive Modify to	char nameToNegister[NUME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking void* socketChat(void *arg){ // find if same conned exist // if exist do send video // intretual; // intretual; // intretual; // if exist do send video // // intretual; // if exist do send video // // intretual; // // intretual; // // // intretual; // // // // intretual; // // // intretual; // // // intretual; // i
Dot Dot	chain.cpp:45: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o cotions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:328: [2] (buffer) char: titically-sized arrays can be improperly restricted, leading to pote erflows or other issues (CWE-119I/CWE-120). Perform bounds o cotions that limit length, or ensure that the size is larger than the ximum possible length. chain.cpp:125: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnp:157: [2] (buffer) memcpy: es not check for buffer overflows when copying to destination (C ke sure destination can always hold the source data. chain.cnp:125: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks coe the opening of special file type (e.g., device files), move thing and to create a race condition, control its ancestors, or change i ments? (CWE-362). chain.cp:121: [2] (misc) open: eck when opening files - can an attacker redirect it (via symlinks coe the opening of special file type (e.g., device files), move thing and to create a race condition, control its ancestors, or change i ments? (CWE-362). chain.cp:125: [1] (buffer) read: eck buffer boundaries if used in a loop including recursive loops WE-120, CWE-20). chain.cpp:122: [1] (buffer) read: eck buffer boundaries if used in a loop including recursive loops WE-120, CWE-20).	ignore-false positive Modify destination length Modify destination length ignore-false positive ignore-false positive ignore-false positive ignore-false positive ignore-false positive Modify to	Char nameToRegister[NUME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking char nameToRegister[NUME_LEN]; //flawfinder_Sverflow : ignore - perform bounds checking lovoid* socketChat(void *arg){

ID-16	Check buffer boundaries if used in a loop including recursive loops	positive			
	(CWE-120, CWE-20).				
	ANALYSIS SUMMARY:				
	Hits = 16				
	Lines analyzed = 3454 in approximately 0.33 seconds (10488 lines/	/second)			
	Physical Source Lines of Code (SLOC) = 2577				
	Hits@level = [0] 56 [1] 5 [2] 10 [3] 0 [4] 1 [5] 0				
	Hits@level+ = [0+] 72 [1+] 16 [2+] 11 [3+] 1 [4+] 1 [5+] 0				
	Hits/KSLOC@level+ = [0+] 27.9395 [1+] 6.20877 [2+] 4.26853 [3+]	0.388048 [4+] 0	388048 [5+] 0		
	Minimum risk level = 1				
	Not associated by the second s				
	Not every hit is necessarily a security vulnerability.				
	You can inhibit a report by adding a comment in this form:				
	// flawfinder: ignore				
	Make *sure* it's a false positive!				
	You can use the optionneverignore to show these.				
	There may be other security vulnerabilities; review your code!				
	See 'Secure Programming HOWTO'				
	(https://dwheeler.com/secure-programs) for more information.				