PF_RING Scan Report

Project Name	PF_RING
Scan Start	Friday, June 21, 2024 12:12:05 PM
Preset	Checkmarx Default
Scan Time	00h:12m:48s
Lines Of Code Scanned	141623
Files Scanned	82
Report Creation Time	Friday, June 21, 2024 12:41:32 PM
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	4/1000 (Vulnerabilities/LOC)
Visibility	Public

Filter Settings

Severity

Included: High, Medium, Low, Information

Excluded: None

Result State

Included: Confirmed, Not Exploitable, To Verify, Urgent, Proposed Not Exploitable

Excluded: None

Assigned to

Included: All

Categories

Included:

Uncategorized	All
Custom	All
PCI DSS v3.2	All
OWASP Top 10 2013	All
FISMA 2014	All
NIST SP 800-53	All
OWASP Top 10 2017	All
OWASP Mobile Top 10 2016	All

Excluded:

Uncategorized	None
Custom	None
PCI DSS v3.2	None
OWASP Top 10 2013	None
FISMA 2014	None

NIST SP 800-53	None
OWASP Top 10 2017	None
OWASP Mobile Top 10 2016	None

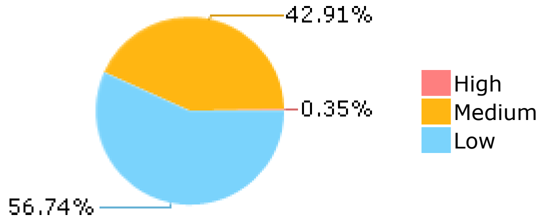
Results Limit

Results limit per query was set to 50

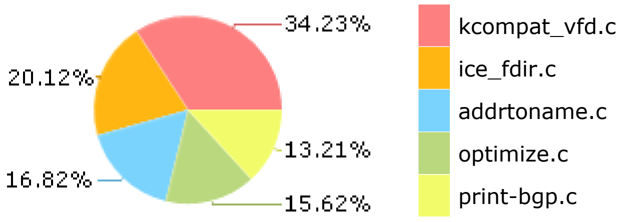
Selected Queries

Selected queries are listed in Result Summary

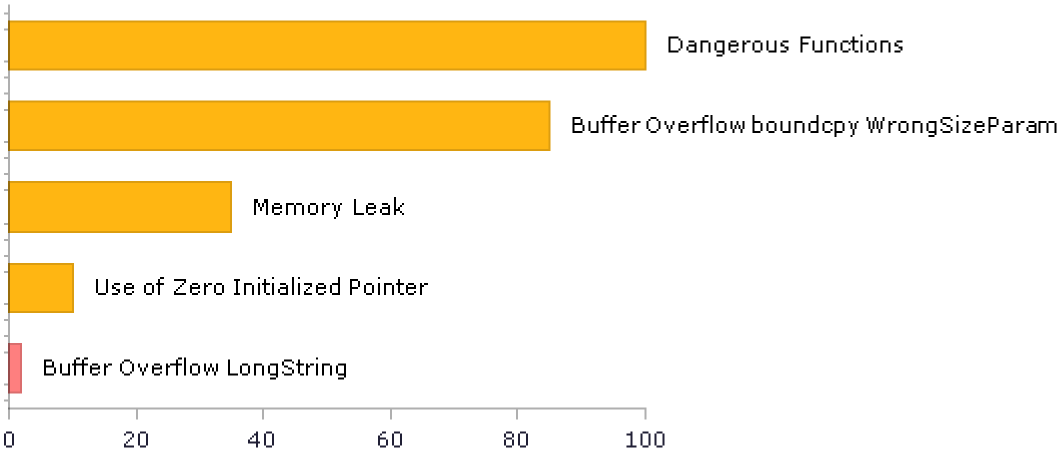
Result Summary



Most Vulnerable Files



Top 5 Vulnerabilities



Scan Summary - OWASP Top 10 2017

Further details and elaboration about vulnerabilities and risks can be found at: OWASP Top 10 2017

Category	Threat Agent	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	App. Specific	EASY	COMMON	EASY	SEVERE	App. Specific	229	101
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	31	31
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	2	1
A4-XML External Entities (XXE)	App. Specific	AVERAGE	COMMON	EASY	SEVERE	App. Specific	0	0
A5-Broken Access Control*	App. Specific	AVERAGE	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A6-Security Misconfiguration	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A7-Cross-Site Scripting (XSS)	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A8-Insecure Deserialization	App. Specific	DIFFICULT	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A9-Using Components with Known Vulnerabilities*	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	MODERATE	App. Specific	100	100
A10-Insufficient Logging & Monitoring	App. Specific	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	App. Specific	0	0

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - OWASP Top 10 2013

Further details and elaboration about vulnerabilities and risks can be found at: OWASP Top 10 2013

Category	Threat Agent	Attack Vectors	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	AVERAGE	SEVERE	ALL DATA	0	0
A2-Broken Authentication and Session Management	EXTERNAL, INTERNAL USERS	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	AFFECTED DATA AND FUNCTIONS	0	0
A3-Cross-Site Scripting (XSS)	EXTERNAL, INTERNAL, ADMIN USERS	AVERAGE	VERY WIDESPREAD	EASY	MODERATE	AFFECTED DATA AND SYSTEM	0	0
A4-Insecure Direct Object References	SYSTEM USERS	EASY	COMMON	EASY	MODERATE	EXPOSED DATA	0	0
A5-Security Misconfiguration	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	EASY	MODERATE	ALL DATA AND SYSTEM	0	0
A6-Sensitive Data Exposure	EXTERNAL, INTERNAL, ADMIN USERS, USERS BROWSERS	DIFFICULT	UNCOMMON	AVERAGE	SEVERE	EXPOSED DATA	0	0
A7-Missing Function Level Access Control*	EXTERNAL, INTERNAL USERS	EASY	COMMON	AVERAGE	MODERATE	EXPOSED DATA AND FUNCTIONS	0	0
A8-Cross-Site Request Forgery (CSRF)	USERS BROWSERS	AVERAGE	COMMON	EASY	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0
A9-Using Components with Known Vulnerabilities*	EXTERNAL USERS, AUTOMATED TOOLS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	100	100
A10-Unvalidated Redirects and Forwards	USERS BROWSERS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0

***** Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - PCI DSS v3.2

Category	Issues Found	Best Fix Locations
PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection	0	0
PCI DSS (3.2) - 6.5.2 - Buffer overflows	89	89
PCI DSS (3.2) - 6.5.3 - Insecure cryptographic storage	0	0
PCI DSS (3.2) - 6.5.4 - Insecure communications	0	0
PCI DSS (3.2) - 6.5.5 - Improper error handling*****	0	0
PCI DSS (3.2) - 6.5.7 - Cross-site scripting (XSS)	0	0
PCI DSS (3.2) - 6.5.8 - Improper access control	0	0
PCI DSS (3.2) - 6.5.9 - Cross-site request forgery	0	0
PCI DSS (3.2) - 6.5.10 - Broken authentication and session management	0	0

***** Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - FISMA 2014

Category	Description	Issues Found	Best Fix Locations
Access Control	Organizations must limit information system access to authorized users, processes acting on behalf of authorized users, or devices (including other information systems) and to the types of transactions and functions that authorized users are permitted to exercise.	2	2
Audit And Accountability*	Organizations must: (i) create, protect, and retain information system audit records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful, unauthorized, or inappropriate information system activity; and (ii) ensure that the actions of individual information system users can be uniquely traced to those users so they can be held accountable for their actions.	1	1
Configuration Management	Organizations must: (i) establish and maintain baseline configurations and inventories of organizational information systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles; and (ii) establish and enforce security configuration settings for information technology products employed in organizational information systems.	4	3
Identification And Authentication*	Organizations must identify information system users, processes acting on behalf of users, or devices and authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	37	30
Media Protection	Organizations must: (i) protect information system media, both paper and digital; (ii) limit access to information on information system media to authorized users; and (iii) sanitize or destroy information system media before disposal or release for reuse.	0	0
System And Communications Protection	Organizations must: (i) monitor, control, and protect organizational communications (i.e., information transmitted or received by organizational information systems) at the external boundaries and key internal boundaries of the information systems; and (ii) employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational information systems.	0	0
System And Information Integrity	Organizations must: (i) identify, report, and correct information and information system flaws in a timely manner; (ii) provide protection from malicious code at appropriate locations within organizational information systems; and (iii) monitor information system security alerts and advisories and take appropriate actions in response.	1	1

***** Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - NIST SP 800-53

Category	Issues Found	Best Fix Locations
AC-12 Session Termination (P2)	0	0
AC-3 Access Enforcement (P1)	33	33
AC-4 Information Flow Enforcement (P1)	0	0
AC-6 Least Privilege (P1)	0	0
AU-9 Protection of Audit Information (P1)	0	0
CM-6 Configuration Settings (P2)	0	0
IA-5 Authenticator Management (P1)	0	0
IA-6 Authenticator Feedback (P2)	0	0
IA-8 Identification and Authentication (Non-Organizational Users) (P1)	0	0
SC-12 Cryptographic Key Establishment and Management (P1)	0	0
SC-13 Cryptographic Protection (P1)	2	1
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	0	0
SC-23 Session Authenticity (P1)*****	8	1
SC-28 Protection of Information at Rest (P1)	0	0
SC-4 Information in Shared Resources (P1)	0	0
SC-5 Denial of Service Protection (P1)*****	193	59
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*****	21	21
SI-11 Error Handling (P2)*****	70	70
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	1	1

***** Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - OWASP Mobile Top 10 2016

Category	Description	Issues Found	Best Fix Locations
M1-Improper Platform Usage	This category covers misuse of a platform feature or failure to use platform security controls. It might include Android intents, platform permissions, misuse of TouchID, the Keychain, or some other security control that is part of the mobile operating system. There are several ways that mobile apps can experience this risk.	0	0
M2-Insecure Data Storage	This category covers insecure data storage and unintended data leakage.	0	0
M3-Insecure Communication	This category covers poor handshaking, incorrect SSL versions, weak negotiation, cleartext communication of sensitive assets, etc.	0	0
M4-Insecure Authentication	This category captures notions of authenticating the end user or bad session management. This can include: -Failing to identify the user at all when that should be required -Failure to maintain the user's identity when it is required -Weaknesses in session management	0	0
M5-Insufficient Cryptography	The code applies cryptography to a sensitive information asset. However, the cryptography is insufficient in some way. Note that anything and everything related to TLS or SSL goes in M3. Also, if the app fails to use cryptography at all when it should, that probably belongs in M2. This category is for issues where cryptography was attempted, but it wasnt done correctly.	0	0
M6-Insecure Authorization	This is a category to capture any failures in authorization (e.g., authorization decisions in the client side, forced browsing, etc.). It is distinct from authentication issues (e.g., device enrolment, user identification, etc.). If the app does not authenticate users at all in a situation where it should (e.g., granting anonymous access to some resource or service when authenticated and authorized access is required), then that is an authentication failure not an authorization failure.	0	0
M7-Client Code Quality	This category is the catch-all for code-level implementation problems in the mobile client. That's distinct from server-side coding mistakes. This would capture things like buffer overflows, format string vulnerabilities, and various other code-level mistakes where the solution is to rewrite some code that's running on the mobile device.	0	0
M8-Code Tampering	This category covers binary patching, local resource modification, method hooking, method swizzling, and dynamic memory modification. Once the application is delivered to the mobile device, the code and data resources are resident there. An attacker can either directly modify the code, change the contents of memory dynamically, change or replace the system APIs that the application uses, or	0	0

	modify the application's data and resources. This can provide the attacker a direct method of subverting the intended use of the software for personal or monetary gain.		
M9-Reverse Engineering	This category includes analysis of the final core binary to determine its source code, libraries, algorithms, and other assets. Software such as IDA Pro, Hopper, otool, and other binary inspection tools give the attacker insight into the inner workings of the application. This may be used to exploit other nascent vulnerabilities in the application, as well as revealing information about back end servers, cryptographic constants and ciphers, and intellectual property.	0	0
M10-Extraneous Functionality	Often, developers include hidden backdoor functionality or other internal development security controls that are not intended to be released into a production environment. For example, a developer may accidentally include a password as a comment in a hybrid app. Another example includes disabling of 2-factor authentication during testing.	0	0

Scan Summary - Custom

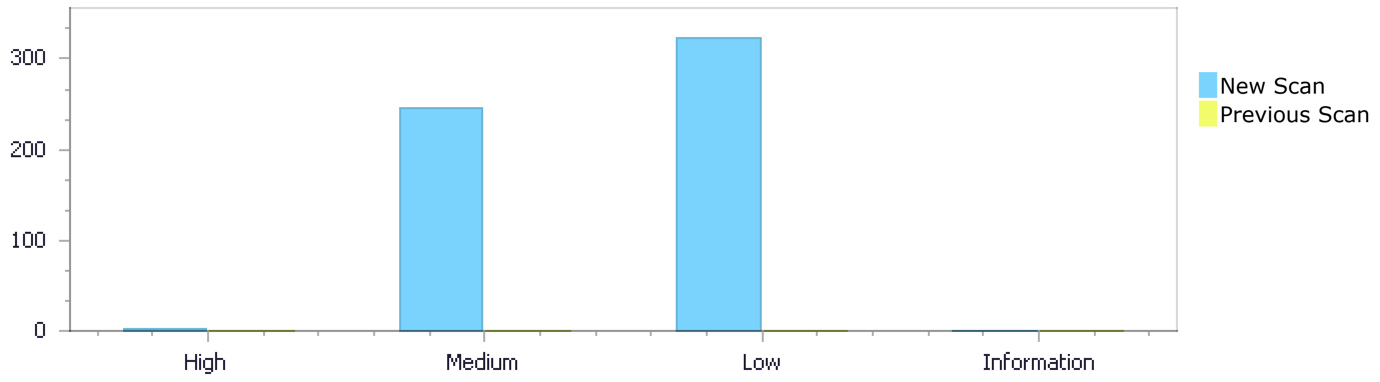
Category	Issues Found	Best Fix Locations
Must audit	0	0
Check	0	0
Optional	0	0

Results Distribution By Status

First scan of the project

	High	Medium	Low	Information	Total
New Issues	2	245	324	0	571
Recurrent Issues	0	0	0	0	0
Total	2	245	324	0	571

Fixed Issues	0	0	0	0	0



Results Distribution By State

	High	Medium	Low	Information	Total
Confirmed	0	0	0	0	0
Not Exploitable	0	0	0	0	0
To Verify	2	245	324	0	571
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	2	245	324	0	571

Result Summary

Vulnerability Type	Occurrences	Severity
Buffer Overflow LongString	2	High
Dangerous Functions	100	Medium
Buffer Overflow boundcpy WrongSizeParam	85	Medium
Memory Leak	35	Medium
Use of Zero Initialized Pointer	10	Medium

Use of Uninitialized Pointer	5	Medium
Wrong Size t Allocation	3	Medium
Inadequate Encryption Strength	2	Medium
Char Overflow	1	Medium
Divide By Zero	1	Medium
Double Free	1	Medium
Integer Overflow	1	Medium
Wrong Memory Allocation	1	Medium
NULL Pointer Dereference	142	Low
Unchecked Return Value	70	Low
Use of Sizeof On a Pointer Type	40	Low
Improper Resource Access Authorization	29	Low
Unchecked Array Index	16	Low
Sizeof Pointer Argument	10	Low
Reliance on DNS Lookups in a Decision	8	Low
TOCTOU	3	Low
Exposure of System Data to Unauthorized Control Sphere	2	Low
Incorrect Permission Assignment For Critical Resources	2	Low
Arithmenic Operation On Boolean	1	Low
Inconsistent Implementations	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
PF_RING/ice_fdir.c	59
PF_RING/addrtoname.c	42
PF_RING/optimize.c	29
PF_RING/i40e_txrx.c	21
PF_RING/kcompat_vfd.c	20
PF_RING/pfutils.c	17
PF_RING/print-bgp.c	10
PF_RING/print-tcp.c	9
PF_RING/print-babel.c	9
PF_RING/print-rx.c	5

Scan Results Details

Buffer Overflow LongString

Query Path:
CPP\Cx\CPP Buffer Overflow\Buffer Overflow LongString Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.2 - Buffer overflows
NIST SP 800-53: SI-10 Information Input Validation (P1)
OWASP Top 10 2017: A1-Injection

Description

Buffer Overflow LongString\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=1
Status	New

The size of the buffer used by create_qos_tc_sysfs in kname, at line 3001 of PF_RING/kcompat_vfd.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that create_qos_tc_sysfs passes to "%d", at line 3001 of PF_RING/kcompat_vfd.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	3010	3012
Object	"%d"	kname

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method static int create_qos_tc_sysfs(struct pci_dev *pdev, struct kobject **tc,

```
....
3010.            int length = snprintf(kname, sizeof(kname), "%d", i);
....
3012.            if (length >= sizeof(kname)) {
```

Buffer Overflow LongString\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=2
Status	New

The size of the buffer used by format_interval in buf, at line 186 of PF_RING/print-hncp.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that format_interval passes to "%u.%0x03us", at line 186 of PF_RING/print-hncp.c, to overwrite the target buffer.

	Source	Destination

File	PF_RING/print-hncp.c	PF_RING/print-hncp.c
Line	191	192
Object	"%u.%0x03us"	buf

Code Snippet
File Name PF_RING/print-hncp.c
Method format_interval(const uint32_t n)

```
....
191.      snprintf(buf[i], sizeof(buf[i]), "%u.%03us", n / 1000, n %
1000);
192.      return buf[i];
```

Dangerous Functions

Query Path:
CPP\Cx\CPP Medium Threat\Dangerous Functions Version:1

Categories

OWASP Top 10 2013: A9-Using Components with Known Vulnerabilities
OWASP Top 10 2017: A9-Using Components with Known Vulnerabilities

Description

Dangerous Functions\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=326
Status	New

The dangerous function, memcpy, was found in use at line 149 in PF_RING/addrtoname.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	164	164
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/addrtoname.c
Method win32_gethostbyaddr(const char *addr, int len, int type)

```
....
164.            memcpy(&addr6.sin6_addr, addr, len);
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=200

	14&pathid=327
Status	New

The dangerous function, memcpy, was found in use at line 279 in PF_RING/addrtoname.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	285	285
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/addrtoname.c
Method ipaddr_string(netdissect_options *ndo, const u_char *ap)

```
....
285.        memcpy(&addr, ap, sizeof(addr));
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=328
Status	New

The dangerous function, memcpy, was found in use at line 338 in PF_RING/addrtoname.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	352	352
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/addrtoname.c
Method ip6addr_string(netdissect_options *ndo, const u_char *ap)

```
....
352.        memcpy(&addr, ap, sizeof(addr));
```

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=329
Status	New

The dangerous function, memcpy, was found in use at line 338 in PF_RING/addrtoname.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	358	358
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/addrtoname.c
Method ip6addr_string(netdissect_options *ndo, const u_char *ap)

```
....
358.        memcpy(p->addr, addr.addr, sizeof(nd_ipv6));
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=330
Status	New

The dangerous function, memcpy, was found in use at line 470 in PF_RING/addrtoname.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	507	507
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/addrtoname.c
Method lookup_bytestring(netdissect_options *ndo, const u_char *bs,

```
....
507.        memcpy(tp->bs_bytes, bs, nlen);
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=331
Status	New

The dangerous function, memcpy, was found in use at line 520 in PF_RING/addrtoname.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	554	554
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/addrtoname.c
Method lookup_nsap(netdissect_options *ndo, const u_char *nsap,

```
....
554.        memcpy((char *)&tp->e_nsap[1], (const char *)nsap,
nsap_length);
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=332
Status	New

The dangerous function, memcpy, was found in use at line 591 in PF_RING/addrtoname.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	612	612
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/addrtoname.c
Method etheraddr_string(netdissect_options *ndo, const uint8_t *ep)

```
....
612.             memcpy (&ea, ep, MAC_ADDR_LEN);
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=333
Status	New

The dangerous function, memcpy, was found in use at line 899 in PF_RING/addrtoname.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination

File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	912	912
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/addrtoname.c
Method init_protoidarray(netdissect_options *ndo)

```
....
912.                memcpy((char *)&protoid[3], (char *)&etype, 2);
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=334
Status	New

The dangerous function, memcpy, was found in use at line 953 in PF_RING/addrtoname.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	990	990
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/addrtoname.c
Method init_etherarray(netdissect_options *ndo)

```
....
990.                memcpy (&ea, el->addr, MAC_ADDR_LEN);
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=335
Status	New

The dangerous function, memcpy, was found in use at line 81 in PF_RING/fttest.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/fttest.c	PF_RING/fttest.c
Line	130	130

Object	memcpy	memcpy

Code Snippet
File Name PF_RING/fttest.c
Method void print_stats() {

```
....
130.     memcpy(&last_time, &end_time, sizeof(last_time));
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=336
Status	New

The dangerous function, memcpy, was found in use at line 124 in PF_RING/i40e_client.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/i40e_client.c	PF_RING/i40e_client.c
Line	143	143
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/i40e_client.c
Method void i40e_notify_client_of_l2_param_changes(struct i40e_vsi *vsi)

```
....
143.         memcpy(&cdev->lan_info.params, &params, sizeof(struct
i40e_params));
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=337
Status	New

The dangerous function, memcpy, was found in use at line 186 in PF_RING/i40e_txrx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	209	209
Object	memcpy	memcpy

Code Snippet	
File Name	PF_RING/i40e_txrx.c
Method	static char *i40e_create_dummy_packet(u8 *dummy_packet, bool ipv4, u8 l4proto,

```
....
209.               memcpy(&ipv6.saddr.in6_u.u6_addr32, data->src_ip6,
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=338
Status	New

The dangerous function, memcpy, was found in use at line 186 in PF_RING/i40e_txrx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	211	211
Object	memcpy	memcpy

Code Snippet	
File Name	PF_RING/i40e_txrx.c
Method	static char *i40e_create_dummy_packet(u8 *dummy_packet, bool ipv4, u8 l4proto,

```
....
211.               memcpy(&ipv6.daddr.in6_u.u6_addr32, data->dst_ip6,
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=339
Status	New

The dangerous function, memcpy, was found in use at line 186 in PF_RING/i40e_txrx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	221	221
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/i40e_txrx.c
Method	static char *i40e_create_dummy_packet(u8 *dummy_packet, bool ipv4, u8 l4proto,

```
....
221.        memcpy(tmp, &eth, sizeof(eth));
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=340
Status	New

The dangerous function, memcpy, was found in use at line 186 in PF_RING/i40e_txrx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	225	225
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/i40e_txrx.c
Method	static char *i40e_create_dummy_packet(u8 *dummy_packet, bool ipv4, u8 l4proto,

```
....
225.                memcpy(tmp, &vlan, sizeof(vlan));
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=341
Status	New

The dangerous function, memcpy, was found in use at line 186 in PF_RING/i40e_txrx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	230	230
Object	memcpy	memcpy

Code Snippet

File Name PF_RING/i40e_txrx.c
Method static char *i40e_create_dummy_packet(u8 *dummy_packet, bool ipv4, u8 l4proto,

```
....
230.                memcpy(tmp, &ip, sizeof(ip));
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=342
Status	New

The dangerous function, memcpy, was found in use at line 186 in PF_RING/i40e_txrx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	233	233
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static char *i40e_create_dummy_packet(u8 *dummy_packet, bool ipv4, u8 l4proto,

```
....
233.                memcpy(tmp, &ipv6, sizeof(ipv6));
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=343
Status	New

The dangerous function, memcpy, was found in use at line 270 in PF_RING/i40e_txrx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	282	282
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/i40e_txrx.c

Method	static void i40e_create_dummy_tcp_packet(u8 *raw_packet, bool ipv4, u8 l4proto,

```
....
282.        memcpy(tcp, tcp_packet, sizeof(tcp_packet));
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=344
Status	New

The dangerous function, memcpy, was found in use at line 3176 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3181	3181
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/ice_fdir.c
Method	static void ice_pkt_insert_ipv6_addr(u8 *pkt, int offset, __be32 *addr)

```
....
3181.           memcpy(pkt + offset + idx * sizeof(*addr), &addr[idx],
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=345
Status	New

The dangerous function, memcpy, was found in use at line 3193 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3198	3198
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/ice_fdir.c
Method	static void ice_pkt_insert_u6_qfi(u8 *pkt, int offset, u8 data)

```
....
3198.       memcpy(pkt + offset, &ret, sizeof(ret));
```

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=346
Status	New

The dangerous function, memcpy, was found in use at line 3207 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3209	3209
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/ice_fdir.c
Method	static void ice_pkt_insert_u8(u8 *pkt, int offset, u8 data)

```
....
3209.       memcpy(pkt + offset, &data, sizeof(data));
```

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=347
Status	New

The dangerous function, memcpy, was found in use at line 3223 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3228	3228
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/ice_fdir.c
Method	static void ice_pkt_insert_u8_tc(u8 *pkt, int offset, u8 data)

```
....
3228.        memcpy(pkt + offset, &high, sizeof(high));
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=348
Status	New

The dangerous function, memcpy, was found in use at line 3223 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3231	3231
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_u8_tc(u8 *pkt, int offset, u8 data)

```
....
3231.        memcpy(pkt + offset + 1, &low, sizeof(low));
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=349
Status	New

The dangerous function, memcpy, was found in use at line 3240 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3242	3242
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_u16(u8 *pkt, int offset, __be16 data)

```
....
3242.        memcpy(pkt + offset, &data, sizeof(data));
```

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=350
Status	New

The dangerous function, memcpy, was found in use at line 3251 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3253	3253
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_u32(u8 *pkt, int offset, __be32 data)

```
....
3253.        memcpy(pkt + offset, &data, sizeof(data));
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=351
Status	New

The dangerous function, memcpy, was found in use at line 3302 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3350	3350
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_get_gen_prgm_pkt(struct ice_hw *hw, struct ice_fdir_fltr *input,

```
....
3350.                memcpy(pkt, ice_fdir_pkt[idx].pkt,
ice_fdir_pkt[idx].pkt_len);
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=352
Status	New

The dangerous function, memcpy, was found in use at line 3302 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3363	3363
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/ice_fdir.c
Method	ice_fdir_get_gen_prgm_pkt(struct ice_hw *hw, struct ice_fdir_fltr *input,

```
....
3363.                       memcpy(pkt, ice_fdir_pkt[idx].tun_pkt,
```

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=353
Status	New

The dangerous function, memcpy, was found in use at line 3302 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3385	3385
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/ice_fdir.c
Method	ice_fdir_get_gen_prgm_pkt(struct ice_hw *hw, struct ice_fdir_fltr *input,

```
....
3385.                     memcpy(pkt, ice_fdir_pkt[idx].tun_pkt,
```

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=354
Status	New

The dangerous function, memcpy, was found in use at line 3302 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3395	3395
Object	memcpy	memcpy

Code Snippet

File Name PF_RING/ice_fdir.c

Method ice_fdir_get_gen_prgm_pkt(struct ice_hw *hw, struct ice_fdir_fltr *input,

```
....
3395.                     memcpy(pkt, ice_fdir_pkt[idx].tun_pkt,
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=355
Status	New

The dangerous function, memcpy, was found in use at line 3302 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3405	3405
Object	memcpy	memcpy

Code Snippet

File Name PF_RING/ice_fdir.c

Method ice_fdir_get_gen_prgm_pkt(struct ice_hw *hw, struct ice_fdir_fltr *input,

```
....
3405.                    memcpy(pkt, ice_fdir_pkt[idx].tun_pkt,
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=356
Status	New

The dangerous function, memcpy, was found in use at line 3302 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3415	3415
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/ice_fdir.c
Method	ice_fdir_get_gen_prgm_pkt(struct ice_hw *hw, struct ice_fdir_fltr *input,

```
....
3415.                    memcpy(pkt, ice_fdir_pkt[idx].tun_pkt,
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=357
Status	New

The dangerous function, memcpy, was found in use at line 3302 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3425	3425
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/ice_fdir.c
Method	ice_fdir_get_gen_prgm_pkt(struct ice_hw *hw, struct ice_fdir_fltr *input,

```
....
3425.                            memcpy(pkt, ice_fdir_pkt[idx].tun_pkt,
```

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=358
Status	New

The dangerous function, memcpy, was found in use at line 3302 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3447	3447
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_get_gen_prgm_pkt(struct ice_hw *hw, struct ice_fdir_fltr *input,

```
....
3447.                            memcpy(pkt, ice_fdir_pkt[idx].tun_pkt,
```

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=359
Status	New

The dangerous function, memcpy, was found in use at line 3302 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3469	3469
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_get_gen_prgm_pkt(struct ice_hw *hw, struct ice_fdir_fltr *input,

```
....
3469.                   memcpy(pkt, ice_fdir_pkt[idx].tun_pkt,
```

Dangerous Functions\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=360
Status	New

The dangerous function, memcpy, was found in use at line 3302 in PF_RING/ice_fdir.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3477	3477
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_get_gen_prgm_pkt(struct ice_hw *hw, struct ice_fdir_fltr *input,

```
....
3477.                   memcpy(pkt, ice_fdir_pkt[idx].tun_pkt,
```

Dangerous Functions\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=361
Status	New

The dangerous function, memcpy, was found in use at line 1478 in PF_RING/optimize.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	1510	1510
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/optimize.c
Method opt_blk(opt_state_t *opt_state, struct block *b, int do_stmts)

```
....
1510.             memcpy((char *)b->val, (char *)p->pred->val, sizeof(b-
>val));
```

Dangerous Functions\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=362
Status	New

The dangerous function, memcpy, was found in use at line 2939 in PF_RING/optimize.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2965	2965
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/optimize.c
Method	install_bpf_program(pcap_t *p, struct bpf_program *fp)

```
....
2965.        memcpy(p->fcode.bf_insns, fp->bf_insns, prog_size);
```

Dangerous Functions\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=363
Status	New

The dangerous function, memcpy, was found in use at line 85 in PF_RING/parsenfsfh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/parsenfsfh.c	PF_RING/parsenfsfh.c
Line	348	348
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/parsenfsfh.c
Method	Parse_fh(netdissect_options *ndo, const unsigned char *fh, u_int len,

```
....
348.                memcpy((char *)fsidp, (const char *)fh, 14);
```

Dangerous Functions\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=364
Status	New

The dangerous function, memcpy, was found in use at line 85 in PF_RING/parsenfsfh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/parsenfsfh.c	PF_RING/parsenfsfh.c
Line	354	354
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/parsenfsfh.c
Method	Parse_fh(netdissect_options *ndo, const unsigned char *fh, u_int len,

```
....
354.                memcpy((char *)tempa, (const char *)fh, 14); /* ensure
alignment */
```

Dangerous Functions\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=365
Status	New

The dangerous function, memcpy, was found in use at line 144 in PF_RING/pfutils.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	199	199
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/pfutils.c
Method	static void forge_udp_packet_fast(u_char *buffer, u_int packet_len, u_int idx) {

```
....
199.       memcpy(matrix_buffer, buffer, sizeof(struct ether_header) +
sizeof(struct compact_ip_hdr) + sizeof(struct compact_udp_hdr));
```

Dangerous Functions\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=366
Status	New

The dangerous function, memcpy, was found in use at line 144 in PF_RING/pfutils.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	201	201
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/pfutils.c
Method	static void forge_udp_packet_fast(u_char *buffer, u_int packet_len, u_int idx) {

```
....
201.       memcpy(buffer, matrix_buffer, sizeof(struct ether_header) +
sizeof(struct compact_ip_hdr) + sizeof(struct compact_udp_hdr));
```

Dangerous Functions\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=367
Status	New

The dangerous function, memcpy, was found in use at line 233 in PF_RING/pfutils.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	247	247
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/pfutils.c
Method	static void forge_udp_packet(u_char *buffer, u_int buffer_len, u_int idx, u_int ip_version) {

```
....
247.     if(reforge_dst_mac) memcpy(buffer, dstmac, 6);
```

Dangerous Functions\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=368
Status	New

The dangerous function, memcpy, was found in use at line 233 in PF_RING/pfutils.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	248	248
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/pfutils.c
Method static void forge_udp_packet(u_char *buffer, u_int buffer_len, u_int idx, u_int ip_version) {

```
....
248.     if(reforge_src_mac) memcpy(&buffer[6], srcmac, 6);
```

Dangerous Functions\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=369
Status	New

The dangerous function, memcpy, was found in use at line 202 in PF_RING/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/print-babel.c	PF_RING/print-babel.c
Line	227	227
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/print-babel.c
Method network_prefix(int ae, int plen, unsigned int omitted,

```
....
227.          memcpy(prefix, v4prefix, 12);
```

Dangerous Functions\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=370
Status	New

The dangerous function, memcpy, was found in use at line 202 in PF_RING/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/print-babel.c	PF_RING/print-babel.c
Line	230	230
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/print-babel.c
Method	network_prefix(int ae, int plen, unsigned int omitted,

```
....
230.             memcpy(prefix, dp, 12 + omitted);
```

Dangerous Functions\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=371
Status	New

The dangerous function, memcpy, was found in use at line 202 in PF_RING/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/print-babel.c	PF_RING/print-babel.c
Line	233	233
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/print-babel.c
Method	network_prefix(int ae, int plen, unsigned int omitted,

```
....
233.            memcpy(prefix + 12 + omitted, p, pb - omitted);
```

Dangerous Functions\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=372
Status	New

The dangerous function, memcpy, was found in use at line 202 in PF_RING/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/print-babel.c	PF_RING/print-babel.c
Line	242	242
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/print-babel.c
Method	network_prefix(int ae, int plen, unsigned int omitted,

```
....
242.            memcpy(prefix, dp, omitted);
```

Dangerous Functions\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=373
Status	New

The dangerous function, memcpy, was found in use at line 202 in PF_RING/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/print-babel.c	PF_RING/print-babel.c
Line	245	245
Object	memcpy	memcpy

Code Snippet

File Name	PF_RING/print-babel.c
Method	network_prefix(int ae, int plen, unsigned int omitted,

```
....
245.                memcpy(prefix + omitted, p, pb - omitted);
```

Dangerous Functions\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=374
Status	New

The dangerous function, memcpy, was found in use at line 202 in PF_RING/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/print-babel.c	PF_RING/print-babel.c
Line	254	254
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/print-babel.c
Method network_prefix(int ae, int plen, unsigned int omitted,

```
....
254.                memcpy(prefix + 8, p, pb - 8);
```

Dangerous Functions\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=375
Status	New

The dangerous function, memcpy, was found in use at line 202 in PF_RING/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	PF_RING/print-babel.c	PF_RING/print-babel.c
Line	262	262
Object	memcpy	memcpy

Code Snippet
File Name PF_RING/print-babel.c
Method network_prefix(int ae, int plen, unsigned int omitted,

```
....
262.      memcpy(p_r, prefix, 16);
```

Buffer Overflow boundcpy WrongSizeParam

Query Path:
CPP\Cx\CPP Buffer Overflow\Buffer Overflow boundcpy WrongSizeParam Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.2 - Buffer overflows
OWASP Top 10 2017: A1-Injection

Description

Buffer Overflow boundcpy WrongSizeParam\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=75
Status	New

The size of the buffer used by ip6addr_string in nd_ipv6, at line 338 of PF_RING/addrtoname.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ip6addr_string passes to nd_ipv6, at line 338 of PF_RING/addrtoname.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	358	358
Object	nd_ipv6	nd_ipv6

Code Snippet
File Name PF_RING/addrtoname.c
Method ip6addr_string(netdissect_options *ndo, const u_char *ap)

```
....
358.        memcpy(p->addr, addr.addr, sizeof(nd_ipv6));
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=76
Status	New

The size of the buffer used by i40e_notify_client_of_l2_param_changes in i40e_params, at line 124 of PF_RING/i40e_client.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that i40e_notify_client_of_l2_param_changes passes to i40e_params, at line 124 of PF_RING/i40e_client.c, to overwrite the target buffer.

	Source	Destination

File	PF_RING/i40e_client.c	PF_RING/i40e_client.c
Line	143	143
Object	i40e_params	i40e_params

Code Snippet
File Name PF_RING/i40e_client.c
Method void i40e_notify_client_of_l2_param_changes(struct i40e_vsi *vsi)

```
....
143.        memcpy(&cdev->lan_info.params, &params, sizeof(struct
i40e_params));
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=77
Status	New

The size of the buffer used by *i40e_create_dummy_packet in eth, at line 186 of PF_RING/i40e_txrx.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *i40e_create_dummy_packet passes to eth, at line 186 of PF_RING/i40e_txrx.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	221	221
Object	eth	eth

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static char *i40e_create_dummy_packet(u8 *dummy_packet, bool ipv4, u8 l4proto,

```
....
221.        memcpy(tmp, &eth, sizeof(eth));
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=78
Status	New

The size of the buffer used by *i40e_create_dummy_packet in vlan, at line 186 of PF_RING/i40e_txrx.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *i40e_create_dummy_packet passes to vlan, at line 186 of PF_RING/i40e_txrx.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	225	225
Object	vlan	vlan

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static char *i40e_create_dummy_packet(u8 *dummy_packet, bool ipv4, u8 l4proto,

```
....
225.                memcpy(tmp, &vlan, sizeof(vlan));
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=79
Status	New

The size of the buffer used by *i40e_create_dummy_packet in ip, at line 186 of PF_RING/i40e_txrx.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *i40e_create_dummy_packet passes to ip, at line 186 of PF_RING/i40e_txrx.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	230	230
Object	ip	ip

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static char *i40e_create_dummy_packet(u8 *dummy_packet, bool ipv4, u8 l4proto,

```
....
230.                memcpy(tmp, &ip, sizeof(ip));
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=80
Status	New

The size of the buffer used by *i40e_create_dummy_packet in ipv6, at line 186 of PF_RING/i40e_txrx.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *i40e_create_dummy_packet passes to ipv6, at line 186 of PF_RING/i40e_txrx.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	233	233
Object	ipv6	ipv6

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static char *i40e_create_dummy_packet(u8 *dummy_packet, bool ipv4, u8 l4proto,

```
....
233.                memcpy(tmp, &ipv6, sizeof(ipv6));
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=81
Status	New

The size of the buffer used by i40e_create_dummy_tcp_packet in tcp_packet, at line 270 of PF_RING/i40e_txrx.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that i40e_create_dummy_tcp_packet passes to tcp_packet, at line 270 of PF_RING/i40e_txrx.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	282	282
Object	tcp_packet	tcp_packet

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static void i40e_create_dummy_tcp_packet(u8 *raw_packet, bool ipv4, u8 l4proto,

```
....
282.          memcpy(tcp, tcp_packet, sizeof(tcp_packet));
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=82
Status	New

The size of the buffer used by ice_pkt_insert_ipv6_addr in addr, at line 3176 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_pkt_insert_ipv6_addr passes to addr, at line 3176 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3182	3182
Object	addr	addr

Code Snippet

File Name	PF_RING/ice_fdir.c
Method	static void ice_pkt_insert_ipv6_addr(u8 *pkt, int offset, __be32 *addr)

```
....
3182.                         sizeof(*addr));
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=83
Status	New

The size of the buffer used by ice_pkt_insert_u6_qfi in ret, at line 3193 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_pkt_insert_u6_qfi passes to ret, at line 3193 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3198	3198
Object	ret	ret

Code Snippet

File Name	PF_RING/ice_fdir.c
Method	static void ice_pkt_insert_u6_qfi(u8 *pkt, int offset, u8 data)

```
....
3198.         memcpy(pkt + offset, &ret, sizeof(ret));
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=84
Status	New

The size of the buffer used by ice_pkt_insert_u8 in data, at line 3207 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_pkt_insert_u8 passes to data, at line 3207 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3209	3209
Object	data	data

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_u8(u8 *pkt, int offset, u8 data)

```
....
3209.        memcpy(pkt + offset, &data, sizeof(data));
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=85
Status	New

The size of the buffer used by ice_pkt_insert_u8_tc in high, at line 3223 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_pkt_insert_u8_tc passes to high, at line 3223 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3228	3228
Object	high	high

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_u8_tc(u8 *pkt, int offset, u8 data)

```
....
3228.        memcpy(pkt + offset, &high, sizeof(high));
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=86
Status	New

The size of the buffer used by ice_pkt_insert_u8_tc in low, at line 3223 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_pkt_insert_u8_tc passes to low, at line 3223 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3231	3231
Object	low	low

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_u8_tc(u8 *pkt, int offset, u8 data)

```
....
3231.         memcpy(pkt + offset + 1, &low, sizeof(low));
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=87
Status	New

The size of the buffer used by ice_pkt_insert_u16 in data, at line 3240 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_pkt_insert_u16 passes to data, at line 3240 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3242	3242
Object	data	data

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_u16(u8 *pkt, int offset, __be16 data)

```
....
3242.         memcpy(pkt + offset, &data, sizeof(data));
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=88
Status	New

The size of the buffer used by ice_pkt_insert_u32 in data, at line 3251 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_pkt_insert_u32 passes to data, at line 3251 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3253	3253
Object	data	data

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_u32(u8 *pkt, int offset, __be32 data)

```
....
3253.       memcpy(pkt + offset, &data, sizeof(data));
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=89
Status	New

The size of the buffer used by opt_blk in ->, at line 1478 of PF_RING/optimize.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that opt_blk passes to ->, at line 1478 of PF_RING/optimize.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	1510	1510
Object	->	->

Code Snippet
File Name PF_RING/optimize.c
Method opt_blk(opt_state_t *opt_state, struct block *b, int do_stmts)

```
....
1510.           memcpy((char *)b->val, (char *)p->pred->val, sizeof(b-
>val));
```

Buffer Overflow boundcpy WrongSizeParam\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=90
Status	New

The size of the buffer used by tcp_verify_signature in Namespace1978044503, at line 892 of PF_RING/print-tcp.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that tcp_verify_signature passes to Namespace1978044503, at line 892 of PF_RING/print-tcp.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/print-tcp.c	PF_RING/print-tcp.c
Line	953	953
Object	Namespace1978044503	Namespace1978044503

Code Snippet
File Name PF_RING/print-tcp.c
Method tcp_verify_signature(netdissect_options *ndo,

```
....
953.          memcpy(tp1.th_sum, &savecsum, sizeof(tp1.th_sum));
```

Buffer Overflow boundcpy WrongSizeParam\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=91
Status	New

The size of the buffer used by i40e_program_fdir_filter in i40e_tx_buffer, at line 101 of PF_RING/i40e_txrx.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that i40e_program_fdir_filter passes to i40e_tx_buffer, at line 101 of PF_RING/i40e_txrx.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	147	147
Object	i40e_tx_buffer	i40e_tx_buffer

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static int i40e_program_fdir_filter(struct i40e_fdir_filter *fdir_data,

```
....
147.        memset(tx_buf, 0, sizeof(struct i40e_tx_buffer));
```

Buffer Overflow boundcpy WrongSizeParam\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=92
Status	New

The size of the buffer used by init_val in opt_state, at line 708 of PF_RING/optimize.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that init_val passes to opt_state, at line 708 of PF_RING/optimize.c, to overwrite the target buffer.

	Source	Destination

File	PF_RING/optimize.c	PF_RING/optimize.c
Line	713	713
Object	opt_state	opt_state

Code Snippet
File Name PF_RING/optimize.c
Method init_val(opt_state_t *opt_state)

```
....
713.        memset((char *)opt_state->hashtbl, 0, sizeof opt_state-
>hashtbl);
```

Buffer Overflow boundcpy WrongSizeParam\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=93
Status	New

The size of the buffer used by opt_deadstores in last, at line 1454 of PF_RING/optimize.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that opt_deadstores passes to last, at line 1454 of PF_RING/optimize.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	1460	1460
Object	last	last

Code Snippet
File Name PF_RING/optimize.c
Method opt_deadstores(opt_state_t *opt_state, register struct block *b)

```
....
1460.       memset((char *)last, 0, sizeof last);
```

Buffer Overflow boundcpy WrongSizeParam\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=94
Status	New

The size of the buffer used by opt_blk in ->, at line 1478 of PF_RING/optimize.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that opt_blk passes to ->, at line 1478 of PF_RING/optimize.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c

Line	1502	1502
Object	->	->

Code Snippet
File Name PF_RING/optimize.c
Method opt_blk(opt_state_t *opt_state, struct block *b, int do_stmts)

```
....
1502.             memset((char *)b->val, 0, sizeof(b->val));
```

Buffer Overflow boundcpy WrongSizeParam\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=95
Status	New

The size of the buffer used by Parse_fh in fsidp, at line 85 of PF_RING/parsenfsfh.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Parse_fh passes to fsidp, at line 85 of PF_RING/parsenfsfh.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/parsenfsfh.c	PF_RING/parsenfsfh.c
Line	346	346
Object	fsidp	fsidp

Code Snippet
File Name PF_RING/parsenfsfh.c
Method Parse_fh(netdissect_options *ndo, const unsigned char *fh, u_int len,

```
....
346.                memset((char *)fsidp, 0, sizeof(*fsidp));
```

Buffer Overflow boundcpy WrongSizeParam\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=96
Status	New

The size of the buffer used by Parse_fh in tempa, at line 85 of PF_RING/parsenfsfh.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Parse_fh passes to tempa, at line 85 of PF_RING/parsenfsfh.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/parsenfsfh.c	PF_RING/parsenfsfh.c
Line	353	353

Object	tempa	tempa

Code Snippet
File Name PF_RING/parsenfsfh.c
Method Parse_fh(netdissect_options *ndo, const unsigned char *fh, u_int len,

```
....
353.                memset((char *)tempa, 0, sizeof(tempa));
```

Buffer Overflow boundcpy WrongSizeParam\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=97
Status	New

The size of the buffer used by tcp_verify_signature in Namespace1978044503, at line 892 of PF_RING/print-tcp.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that tcp_verify_signature passes to Namespace1978044503, at line 892 of PF_RING/print-tcp.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/print-tcp.c	PF_RING/print-tcp.c
Line	951	951
Object	Namespace1978044503	Namespace1978044503

Code Snippet
File Name PF_RING/print-tcp.c
Method tcp_verify_signature(netdissect_options *ndo,

```
....
951.            memset(tp1.th_sum, 0, sizeof(tp1.th_sum));
```

Buffer Overflow boundcpy WrongSizeParam\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=98
Status	New

The size of the buffer used by ice_fdir_comp_rules_basic in ->, at line 4203 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_fdir_comp_rules_basic passes to ->, at line 4203 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4207	4207

Object	->	->

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_comp_rules_basic(struct ice_fdir_fltr *a, struct ice_fdir_fltr *b)

```
....
4207.        if (memcmp(&a->ip, &b->ip, sizeof(a->ip)))
```

Buffer Overflow boundcpy WrongSizeParam\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=99
Status	New

The size of the buffer used by ice_fdir_comp_rules_basic in ->, at line 4203 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_fdir_comp_rules_basic passes to ->, at line 4203 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4209	4209
Object	->	->

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_comp_rules_basic(struct ice_fdir_fltr *a, struct ice_fdir_fltr *b)

```
....
4209.        if (memcmp(&a->mask, &b->mask, sizeof(a->mask)))
```

Buffer Overflow boundcpy WrongSizeParam\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=100
Status	New

The size of the buffer used by ice_fdir_comp_rules_extended in ->, at line 4223 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_fdir_comp_rules_extended passes to ->, at line 4223 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4228	4228

Object	->	->

Code Snippet

File Name PF_RING/ice_fdir.c

Method ice_fdir_comp_rules_extended(struct ice_fdir_fltr *a, struct ice_fdir_fltr *b)

```
....
4228.         if (memcmp(&a->gtpu_data, &b->gtpu_data, sizeof(a-
>gtpu_data)))
```

Buffer Overflow boundcpy WrongSizeParam\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=101
Status	New

The size of the buffer used by ice_fdir_comp_rules_extended in ->, at line 4223 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_fdir_comp_rules_extended passes to ->, at line 4223 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4230	4230
Object	->	->

Code Snippet

File Name PF_RING/ice_fdir.c

Method ice_fdir_comp_rules_extended(struct ice_fdir_fltr *a, struct ice_fdir_fltr *b)

```
....
4230.         if (memcmp(&a->gtpu_mask, &b->gtpu_mask, sizeof(a-
>gtpu_mask)))
```

Buffer Overflow boundcpy WrongSizeParam\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=102
Status	New

The size of the buffer used by ice_fdir_comp_rules_extended in ->, at line 4223 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_fdir_comp_rules_extended passes to ->, at line 4223 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c

Line	4232	4232
Object	->	->

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_comp_rules_extended(struct ice_fdir_fltr *a, struct ice_fdir_fltr *b)

```
....
4232.        if (memcmp(&a->l2tpv3_data, &b->l2tpv3_data, sizeof(a-
>l2tpv3_data)))
```

Buffer Overflow boundcpy WrongSizeParam\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=103
Status	New

The size of the buffer used by ice_fdir_comp_rules_extended in ->, at line 4223 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_fdir_comp_rules_extended passes to ->, at line 4223 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4234	4234
Object	->	->

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_comp_rules_extended(struct ice_fdir_fltr *a, struct ice_fdir_fltr *b)

```
....
4234.        if (memcmp(&a->l2tpv3_mask, &b->l2tpv3_mask, sizeof(a-
>l2tpv3_mask)))
```

Buffer Overflow boundcpy WrongSizeParam\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=104
Status	New

The size of the buffer used by ice_fdir_comp_rules_extended in ->, at line 4223 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_fdir_comp_rules_extended passes to ->, at line 4223 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination

File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4236	4236
Object	->	->

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_comp_rules_extended(struct ice_fdir_fltr *a, struct ice_fdir_fltr *b)

```
....
4236.       if (memcmp(&a->ext_data, &b->ext_data, sizeof(a->ext_data)))
```

Buffer Overflow boundcpy WrongSizeParam\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=105
Status	New

The size of the buffer used by ice_fdir_comp_rules_extended in ->, at line 4223 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_fdir_comp_rules_extended passes to ->, at line 4223 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4238	4238
Object	->	->

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_comp_rules_extended(struct ice_fdir_fltr *a, struct ice_fdir_fltr *b)

```
....
4238.       if (memcmp(&a->ext_mask, &b->ext_mask, sizeof(a->ext_mask)))
```

Buffer Overflow boundcpy WrongSizeParam\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=106
Status	New

The size of the buffer used by ice_fdir_comp_rules_extended in ->, at line 4223 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_fdir_comp_rules_extended passes to ->, at line 4223 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination

File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4240	4240
Object	->	->

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_comp_rules_extended(struct ice_fdir_fltr *a, struct ice_fdir_fltr *b)

```
....
4240.        if (memcmp(&a->ecpri_data, &b->ecpri_data, sizeof(a-
>ecpri_data)))
```

Buffer Overflow boundcpy WrongSizeParam\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=107
Status	New

The size of the buffer used by ice_fdir_comp_rules_extended in ->, at line 4223 of PF_RING/ice_fdir.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ice_fdir_comp_rules_extended passes to ->, at line 4223 of PF_RING/ice_fdir.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4242	4242
Object	->	->

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_comp_rules_extended(struct ice_fdir_fltr *a, struct ice_fdir_fltr *b)

```
....
4242.        if (memcmp(&a->ecpri_mask, &b->ecpri_mask, sizeof(a-
>ecpri_mask)))
```

Buffer Overflow boundcpy WrongSizeParam\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=108
Status	New

The size of the buffer used by *i40e_create_dummy_packet in __be32, at line 186 of PF_RING/i40e_txrx.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *i40e_create_dummy_packet passes to __be32, at line 186 of PF_RING/i40e_txrx.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	210	210
Object	__be32	__be32

Code Snippet

File Name	PF_RING/i40e_txrx.c
Method	static char *i40e_create_dummy_packet(u8 *dummy_packet, bool ipv4, u8 l4proto,

```
....
210.                           sizeof(__be32) * 4);
```

Buffer Overflow boundcpy WrongSizeParam\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=109
Status	New

The size of the buffer used by *i40e_create_dummy_packet in __be32, at line 186 of PF_RING/i40e_txrx.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *i40e_create_dummy_packet passes to __be32, at line 186 of PF_RING/i40e_txrx.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	212	212
Object	__be32	__be32

Code Snippet

File Name	PF_RING/i40e_txrx.c
Method	static char *i40e_create_dummy_packet(u8 *dummy_packet, bool ipv4, u8 l4proto,

```
....
212.                           sizeof(__be32) * 4);
```

Buffer Overflow boundcpy WrongSizeParam\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=110
Status	New

The size of the buffer used by forge_udp_packet_fast in compact_udp_hdr, at line 144 of PF_RING/pfutils.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that forge_udp_packet_fast passes to compact_udp_hdr, at line 144 of PF_RING/pfutils.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	199	199
Object	compact_udp_hdr	compact_udp_hdr

Code Snippet

File Name	PF_RING/pfutils.c
Method	static void forge_udp_packet_fast(u_char *buffer, u_int packet_len, u_int idx) {

```
....
199.        memcpy(matrix_buffer, buffer, sizeof(struct ether_header) +
sizeof(struct compact_ip_hdr) + sizeof(struct compact_udp_hdr));
```

Buffer Overflow boundcpy WrongSizeParam\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=111
Status	New

The size of the buffer used by forge_udp_packet_fast in ether_header, at line 144 of PF_RING/pfutils.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that forge_udp_packet_fast passes to ether_header, at line 144 of PF_RING/pfutils.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	199	199
Object	ether_header	ether_header

Code Snippet

File Name	PF_RING/pfutils.c
Method	static void forge_udp_packet_fast(u_char *buffer, u_int packet_len, u_int idx) {

```
....
199.        memcpy(matrix_buffer, buffer, sizeof(struct ether_header) +
sizeof(struct compact_ip_hdr) + sizeof(struct compact_udp_hdr));
```

Buffer Overflow boundcpy WrongSizeParam\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=112
Status	New

The size of the buffer used by forge_udp_packet_fast in compact_ip_hdr, at line 144 of PF_RING/pfutils.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that forge_udp_packet_fast passes to compact_ip_hdr, at line 144 of PF_RING/pfutils.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	199	199
Object	compact_ip_hdr	compact_ip_hdr

Code Snippet
File Name PF_RING/pfutils.c
Method static void forge_udp_packet_fast(u_char *buffer, u_int packet_len, u_int idx) {

```
....
199.      memcpy(matrix_buffer, buffer, sizeof(struct ether_header) +
sizeof(struct compact_ip_hdr) + sizeof(struct compact_udp_hdr));
```

Buffer Overflow boundcpy WrongSizeParam\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=113
Status	New

The size of the buffer used by forge_udp_packet_fast in compact_udp_hdr, at line 144 of PF_RING/pfutils.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that forge_udp_packet_fast passes to compact_udp_hdr, at line 144 of PF_RING/pfutils.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	201	201
Object	compact_udp_hdr	compact_udp_hdr

Code Snippet
File Name PF_RING/pfutils.c
Method static void forge_udp_packet_fast(u_char *buffer, u_int packet_len, u_int idx) {

```
....
201.      memcpy(buffer, matrix_buffer, sizeof(struct ether_header) +
sizeof(struct compact_ip_hdr) + sizeof(struct compact_udp_hdr));
```

Buffer Overflow boundcpy WrongSizeParam\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=114
Status	New

The size of the buffer used by forge_udp_packet_fast in ether_header, at line 144 of PF_RING/pfutils.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that forge_udp_packet_fast passes to ether_header, at line 144 of PF_RING/pfutils.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	201	201
Object	ether_header	ether_header

Code Snippet
File Name PF_RING/pfutils.c
Method static void forge_udp_packet_fast(u_char *buffer, u_int packet_len, u_int idx) {

```
....
201.       memcpy(buffer, matrix_buffer, sizeof(struct ether_header) +
sizeof(struct compact_ip_hdr) + sizeof(struct compact_udp_hdr));
```

Buffer Overflow boundcpy WrongSizeParam\Path 41:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=115
Status New

The size of the buffer used by forge_udp_packet_fast in compact_ip_hdr, at line 144 of PF_RING/pfutils.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that forge_udp_packet_fast passes to compact_ip_hdr, at line 144 of PF_RING/pfutils.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	201	201
Object	compact_ip_hdr	compact_ip_hdr

Code Snippet
File Name PF_RING/pfutils.c
Method static void forge_udp_packet_fast(u_char *buffer, u_int packet_len, u_int idx) {

```
....
201.       memcpy(buffer, matrix_buffer, sizeof(struct ether_header) +
sizeof(struct compact_ip_hdr) + sizeof(struct compact_udp_hdr));
```

Buffer Overflow boundcpy WrongSizeParam\Path 42:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=116

Status	New

The size of the buffer used by find_levels in opt_state, at line 406 of PF_RING/optimize.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that find_levels passes to opt_state, at line 406 of PF_RING/optimize.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	408	408
Object	opt_state	opt_state

Code Snippet
File Name PF_RING/optimize.c
Method find_levels(opt_state_t *opt_state, struct icode *ic)

```
....
408.        memset((char *)opt_state->levels, 0, opt_state->n_blocks *
sizeof(*opt_state->levels));
```

Buffer Overflow boundcpy WrongSizeParam\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=117
Status	New

The size of the buffer used by find_levels in opt_state, at line 406 of PF_RING/optimize.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that find_levels passes to opt_state, at line 406 of PF_RING/optimize.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	408	408
Object	opt_state	opt_state

Code Snippet
File Name PF_RING/optimize.c
Method find_levels(opt_state_t *opt_state, struct icode *ic)

```
....
408.        memset((char *)opt_state->levels, 0, opt_state->n_blocks *
sizeof(*opt_state->levels));
```

Buffer Overflow boundcpy WrongSizeParam\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=118
Status	New

The size of the buffer used by find_edom in opt_state, at line 470 of PF_RING/optimize.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that find_edom passes to opt_state, at line 470 of PF_RING/optimize.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	487	487
Object	opt_state	opt_state

Code Snippet
File Name PF_RING/optimize.c
Method find_edom(opt_state_t *opt_state, struct block *root)

```
....
487.        memset(root->et.edom, 0, opt_state->edgewords *
sizeof(*(uset)0));
```

Buffer Overflow boundcpy WrongSizeParam\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=119
Status	New

The size of the buffer used by find_edom in opt_state, at line 470 of PF_RING/optimize.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that find_edom passes to opt_state, at line 470 of PF_RING/optimize.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	488	488
Object	opt_state	opt_state

Code Snippet
File Name PF_RING/optimize.c
Method find_edom(opt_state_t *opt_state, struct block *root)

```
....
488.        memset(root->ef.edom, 0, opt_state->edgewords *
sizeof(*(uset)0));
```

Buffer Overflow boundcpy WrongSizeParam\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=120
Status	New

The size of the buffer used by find_closure in opt_state, at line 505 of PF_RING/optimize.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that find_closure passes to opt_state, at line 505 of PF_RING/optimize.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	514	514
Object	opt_state	opt_state

Code Snippet
File Name PF_RING/optimize.c
Method find_closure(opt_state_t *opt_state, struct block *root)

```
....
514.                opt_state->n_blocks * opt_state->nodewords *
sizeof(*opt_state->all_closure_sets));
```

Buffer Overflow boundcpy WrongSizeParam\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=121
Status	New

The size of the buffer used by find_closure in opt_state, at line 505 of PF_RING/optimize.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that find_closure passes to opt_state, at line 505 of PF_RING/optimize.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	514	514
Object	opt_state	opt_state

Code Snippet
File Name PF_RING/optimize.c
Method find_closure(opt_state_t *opt_state, struct block *root)

```
....
514.                opt_state->n_blocks * opt_state->nodewords *
sizeof(*opt_state->all_closure_sets));
```

Buffer Overflow boundcpy WrongSizeParam\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=122
Status	New

The size of the buffer used by find_closure in opt_state, at line 505 of PF_RING/optimize.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that find_closure passes to opt_state, at line 505 of PF_RING/optimize.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	514	514
Object	opt_state	opt_state

Code Snippet

File Name	PF_RING/optimize.c
Method	find_closure(opt_state_t *opt_state, struct block *root)

```
....
514.                opt_state->n_blocks * opt_state->nodewords *
sizeof(*opt_state->all_closure_sets));
```

Buffer Overflow boundcpy WrongSizeParam\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=123
Status	New

The size of the buffer used by init_val in opt_state, at line 708 of PF_RING/optimize.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that init_val passes to opt_state, at line 708 of PF_RING/optimize.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	712	712
Object	opt_state	opt_state

Code Snippet

File Name	PF_RING/optimize.c
Method	init_val(opt_state_t *opt_state)

```
....
712.        memset((char *)opt_state->vmap, 0, opt_state->maxval *
sizeof(*opt_state->vmap));
```

Buffer Overflow boundcpy WrongSizeParam\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=124
Status	New

The size of the buffer used by init_val in opt_state, at line 708 of PF_RING/optimize.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that init_val passes to opt_state, at line 708 of PF_RING/optimize.c, to overwrite the target buffer.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	712	712
Object	opt_state	opt_state

Code Snippet
File Name PF_RING/optimize.c
Method init_val(opt_state_t *opt_state)

```
....
712.        memset((char *)opt_state->vmap, 0, opt_state->maxval *
sizeof(*opt_state->vmap));
```

Memory Leak

Query Path:
CPP\Cx\CPP Medium Threat\Memory Leak Version:1

Categories

NIST SP 800-53: SC-5 Denial of Service Protection (P1)

Description

Memory Leak\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=427
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	313	313
Object	name	name

Code Snippet
File Name PF_RING/addrtoname.c
Method ipaddr_string(netdissect_options *ndo, const u_char *ap)

```
....
313.                    p->name = strdup(hp->h_name);
```

Memory Leak\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=200

	14&pathid=428
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	326	326
Object	name	name

Code Snippet

File Name PF_RING/addrtoname.c

Method ipaddr_string(netdissect_options *ndo, const u_char *ap)

```
....
326.        p->name = strdup(intoa(addr));
```

Memory Leak\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=429
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	376	376
Object	name	name

Code Snippet

File Name PF_RING/addrtoname.c

Method ip6addr_string(netdissect_options *ndo, const u_char *ap)

```
....
376.                p->name = strdup(hp->h_name);
```

Memory Leak\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=430
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	390	390

Object	name	name

Code Snippet
File Name PF_RING/addrtoname.c
Method ip6addr_string(netdissect_options *ndo, const u_char *ap)

```
....
390.        p->name = strdup(cp);
```

Memory Leak\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=431
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	457	457
Object	e_nxt	e_nxt

Code Snippet
File Name PF_RING/addrtoname.c
Method lookup_emem(netdissect_options *ndo, const u_char *ep)

```
....
457.        tp->e_nxt = (struct enamemem *)calloc(1, sizeof(*tp));
```

Memory Leak\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=432
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	502	502
Object	bs_bytes	bs_bytes

Code Snippet
File Name PF_RING/addrtoname.c
Method lookup_bytestring(netdissect_options *ndo, const u_char *bs,

```
....
502.          tp->bs_bytes = (u_char *) calloc(1, nlen);
```

Memory Leak\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=433
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	509	509
Object	bs_nxt	bs_nxt

Code Snippet

File Name	PF_RING/addrtoname.c
Method	lookup_bytestring(netdissect_options *ndo, const u_char *bs,

```
....
509.          tp->bs_nxt = (struct bsnamemem *)calloc(1, sizeof(*tp));
```

Memory Leak\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=434
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	550	550
Object	e_nsap	e_nsap

Code Snippet

File Name	PF_RING/addrtoname.c
Method	lookup_nsap(netdissect_options *ndo, const u_char *nsap,

```
....
550.          tp->e_nsap = (u_char *)malloc(nsap_length + 1);
```

Memory Leak\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=435
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	555	555
Object	e_nxt	e_nxt

Code Snippet
File Name PF_RING/addrtoname.c
Method lookup_nsap(netdissect_options *ndo, const u_char *nsap,

```
....
555.        tp->e_nxt = (struct enamemem *)calloc(1, sizeof(*tp));
```

Memory Leak\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=436
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	583	583
Object	p_nxt	p_nxt

Code Snippet
File Name PF_RING/addrtoname.c
Method lookup_protoid(netdissect_options *ndo, const u_char *pi)

```
....
583.        tp->p_nxt = (struct protoidmem *)calloc(1, sizeof(*tp));
```

Memory Leak\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=437
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	614	614

Object	e_name	e_name

Code Snippet
File Name PF_RING/addrtoname.c
Method etheraddr_string(netdissect_options *ndo, const uint8_t *ep)

```
....
614.                        tp->e_name = strdup(buf2);
```

Memory Leak\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=438
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	635	635
Object	e_name	e_name

Code Snippet
File Name PF_RING/addrtoname.c
Method etheraddr_string(netdissect_options *ndo, const uint8_t *ep)

```
....
635.            tp->e_name = strdup(buf);
```

Memory Leak\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=439
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	664	664
Object	bs_name	bs_name

Code Snippet
File Name PF_RING/addrtoname.c
Method le64addr_string(netdissect_options *ndo, const uint8_t *ep)

```
....
664.         tp->bs_name = strdup(buf);
```

Memory Leak\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=440
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	693	693
Object	cp	cp

Code Snippet
File Name PF_RING/addrtoname.c
Method linkaddr_string(netdissect_options *ndo, const uint8_t *ep,

```
....
693.         tp->bs_name = cp = (char *)malloc(len*3);
```

Memory Leak\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=441
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	722	722
Object	cp	cp

Code Snippet
File Name PF_RING/addrtoname.c
Method isonsap_string(netdissect_options *ndo, const uint8_t *nsap,

```
....
722.         tp->e_name = cp = (char
*)malloc(sizeof("xx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx"));
```

Memory Leak\Path 16:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=442
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	753	753
Object	name	name

Code Snippet
File Name PF_RING/addrtoname.c
Method tcpport_string(netdissect_options *ndo, u_short port)

```
....
753.         tp->name = strdup(buf);
```

Memory Leak\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=443
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	775	775
Object	name	name

Code Snippet
File Name PF_RING/addrtoname.c
Method udpport_string(netdissect_options *ndo, u_short port)

```
....
775.         tp->name = strdup(buf);
```

Memory Leak\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=444
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c

Line	804	804
Object	name	name

Code Snippet
File Name PF_RING/addrtoname.c
Method ipxsap_string(netdissect_options *ndo, u_short port)

```
....
804.         tp->name = strdup(buf);
```

Memory Leak\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=445
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	833	833
Object	name	name

Code Snippet
File Name PF_RING/addrtoname.c
Method init_servarray(netdissect_options *ndo)

```
....
833.                     table->name = strdup(buf);
```

Memory Leak\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=446
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	835	835
Object	name	name

Code Snippet
File Name PF_RING/addrtoname.c
Method init_servarray(netdissect_options *ndo)

```
....
835.                table->name = strdup(sv->s_name);
```

Memory Leak\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=447
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	914	914
Object	p_name	p_name

Code Snippet
File Name PF_RING/addrtoname.c
Method init_protoidarray(netdissect_options *ndo)

```
....
914.            tp->p_name = strdup(eproto_db[i].s);
```

Memory Leak\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=448
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	992	992
Object	e_name	e_name

Code Snippet
File Name PF_RING/addrtoname.c
Method init_etherarray(netdissect_options *ndo)

```
....
992.                tp->e_name = strdup(name);
```

Memory Leak\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=449
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	1293	1293
Object	ptr	ptr

Code Snippet
File Name PF_RING/addrtoname.c
Method newhnamemem(netdissect_options *ndo)

```
....
1293.              ptr = (struct hnamemem *)calloc(num, sizeof (*ptr));
```

Memory Leak\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=450
Status	New

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	1313	1313
Object	ptr	ptr

Code Snippet
File Name PF_RING/addrtoname.c
Method newh6namemem(netdissect_options *ndo)

```
....
1313.              ptr = (struct h6namemem *)calloc(num, sizeof (*ptr));
```

Memory Leak\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=451
Status	New

	Source	Destination
File	PF_RING/fttest.c	PF_RING/fttest.c
Line	313	313

Object	pulse_timestamp	pulse_timestamp

Code Snippet

File Name	PF_RING/fttest.c
Method	int main(int argc, char* argv[]) {

```
....
313.     pulse_timestamp = calloc(CACHE_LINE_LEN/sizeof(u_int64_t),
sizeof(u_int64_t));
```

Memory Leak\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=452
Status	New

	Source	Destination
File	PF_RING/fttest.c	PF_RING/fttest.c
Line	218	218
Object	forged_packets	forged_packets

Code Snippet

File Name	PF_RING/fttest.c
Method	void packet_consumer() {

```
....
218.     forged_packets = (u_char *) calloc(num_forged_packets,
packet_len);
```

Memory Leak\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=453
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2529	2529
Object	blocks	blocks

Code Snippet

File Name	PF_RING/optimize.c
Method	opt_init(opt_state_t *opt_state, struct icode *ic)

```
....
2529.       opt_state->blocks = (struct block **)calloc(n,
sizeof(*opt_state->blocks));
```

Memory Leak\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=454
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2549	2549
Object	edges	edges

Code Snippet
File Name PF_RING/optimize.c
Method opt_init(opt_state_t *opt_state, struct icode *ic)

```
....
2549.       opt_state->edges = (struct edge **)calloc(opt_state-
>n_edges, sizeof(*opt_state->edges));
```

Memory Leak\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=455
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2557	2557
Object	levels	levels

Code Snippet
File Name PF_RING/optimize.c
Method opt_init(opt_state_t *opt_state, struct icode *ic)

```
....
2557.       opt_state->levels = (struct block **)calloc(opt_state-
>n_blocks, sizeof(*opt_state->levels));
```

Memory Leak\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=456
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2617	2617
Object	space	space

Code Snippet

File Name	PF_RING/optimize.c
Method	opt_init(opt_state_t *opt_state, struct icode *ic)

```
....
2617.       opt_state->space = (bpf_u_int32 *)malloc(block_memsize +
edge_memsize);
```

Memory Leak\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=457
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2656	2656
Object	vmap	vmap

Code Snippet

File Name	PF_RING/optimize.c
Method	opt_init(opt_state_t *opt_state, struct icode *ic)

```
....
2656.       opt_state->vmap = (struct vmapinfo *)calloc(opt_state-
>maxval, sizeof(*opt_state->vmap));
```

Memory Leak\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=458
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2660	2660
Object	vnode_base	vnode_base

Code Snippet
File Name PF_RING/optimize.c
Method opt_init(opt_state_t *opt_state, struct icode *ic)

```
....
2660.       opt_state->vnode_base = (struct valnode *)calloc(opt_state->maxval, sizeof(*opt_state->vnode_base));
```

Memory Leak\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=459
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2959	2959
Object	bf_insns	bf_insns

Code Snippet
File Name PF_RING/optimize.c
Method install_bpf_program(pcap_t *p, struct bpf_program *fp)

```
....
2959.       p->fcode.bf_insns = (struct bpf_insn *)malloc(prog_size);
```

Memory Leak\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=460
Status	New

	Source	Destination
File	PF_RING/print-tcp.c	PF_RING/print-tcp.c
Line	297	297
Object	nxt	nxt

Code Snippet	
File Name	PF_RING/print-tcp.c
Method	tcp_print(netdissect_options *ndo,

```
....
297.                                               th->nxt = (struct
tcp_seq_hash6 *)
```

Memory Leak\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=461
Status	New

	Source	Destination
File	PF_RING/print-tcp.c	PF_RING/print-tcp.c
Line	355	355
Object	nxt	nxt

Code Snippet	
File Name	PF_RING/print-tcp.c
Method	tcp_print(netdissect_options *ndo,

```
....
355.                                               th->nxt = (struct
tcp_seq_hash *)
```

Use of Zero Initialized Pointer

Query Path:

CPP\Cx\CPP Medium Threat\Use of Zero Initialized Pointer Version:1

Categories

NIST SP 800-53: SC-5 Denial of Service Protection (P1)

Description

Use of Zero Initialized Pointer\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=467
Status	New

The variable declared in ptr at PF_RING/addrtoname.c in line 1285 is not initialized when it is used by ptr at PF_RING/addrtoname.c in line 1285.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c

Line	1288	1299
Object	ptr	ptr

Code Snippet
File Name PF_RING/addrtoname.c
Method newhnamemem(netdissect_options *ndo)

```
....
1288.       static struct hnamemem *ptr = NULL;
....
1299.       p = ptr++;
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=468
Status	New

The variable declared in ptr at PF_RING/addrtoname.c in line 1285 is not initialized when it is used by ptr at PF_RING/addrtoname.c in line 1285.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	1288	1293
Object	ptr	ptr

Code Snippet
File Name PF_RING/addrtoname.c
Method newhnamemem(netdissect_options *ndo)

```
....
1288.       static struct hnamemem *ptr = NULL;
....
1293.             ptr = (struct hnamemem *)calloc(num, sizeof (*ptr));
```

Use of Zero Initialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=469
Status	New

The variable declared in ptr at PF_RING/addrtoname.c in line 1305 is not initialized when it is used by ptr at PF_RING/addrtoname.c in line 1305.

	Source	Destination

File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	1308	1319
Object	ptr	ptr

Code Snippet
File Name PF_RING/addrtoname.c
Method newh6namemem(netdissect_options *ndo)

```
....
1308.       static struct h6namemem *ptr = NULL;
....
1319.       p = ptr++;
```

Use of Zero Initialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=470
Status	New

The variable declared in ptr at PF_RING/addrtoname.c in line 1305 is not initialized when it is used by ptr at PF_RING/addrtoname.c in line 1305.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	1308	1313
Object	ptr	ptr

Code Snippet
File Name PF_RING/addrtoname.c
Method newh6namemem(netdissect_options *ndo)

```
....
1308.       static struct h6namemem *ptr = NULL;
....
1313.             ptr = (struct h6namemem *)calloc(num, sizeof (*ptr));
```

Use of Zero Initialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=471
Status	New

The variable declared in cdev at PF_RING/i40e_client.c in line 347 is not initialized when it is used by cdev at PF_RING/i40e_client.c in line 347.

	Source	Destination
File	PF_RING/i40e_client.c	PF_RING/i40e_client.c
Line	349	354
Object	cdev	cdev

Code Snippet
File Name PF_RING/i40e_client.c
Method static void i40e_client_add_instance(struct i40e_pf *pf)

```
....
349.         struct i40e_client_instance *cdev = NULL;
....
354.         cdev = kzalloc(sizeof(*cdev), GFP_KERNEL);
```

Use of Zero Initialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=472
Status	New

The variable declared in rp at PF_RING/print-domain.c in line 185 is not initialized when it is used by rp at PF_RING/print-domain.c in line 185.

	Source	Destination
File	PF_RING/print-domain.c	PF_RING/print-domain.c
Line	189	285
Object	rp	rp

Code Snippet
File Name PF_RING/print-domain.c
Method fqdn_print(netdissect_options *ndo,

```
....
189.         const u_char *rp = NULL;
....
285.                           rp += 1 + 1;
```

Use of Zero Initialized Pointer\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=473
Status	New

The variable declared in skb at PF_RING/i40e_txrx.c in line 2800 is not initialized when it is used by skb at PF_RING/i40e_txrx.c in line 2800.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	2966	2973
Object	skb	skb

Code Snippet

File Name	PF_RING/i40e_txrx.c
Method	static int i40e_clean_rx_irq(struct i40e_ring *rx_ring, int budget)

```
....
2966.                   skb = NULL;
....
2973.           rx_ring->skb = skb;
```

Use of Zero Initialized Pointer\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=474
Status	New

The variable declared in adev at PF_RING/ice_idc.c in line 582 is not initialized when it is used by cdev_infos at PF_RING/ice_idc.c in line 670.

	Source	Destination
File	PF_RING/ice_idc.c	PF_RING/ice_idc.c
Line	632	691
Object	adev	cdev_infos

Code Snippet

File Name	PF_RING/ice_idc.c
Method	int ice_plug_aux_dev(struct iidc_core_dev_info *cdev_info, const char *name)

```
....
632.                    cdev_info->adev = NULL;
```

File Name	PF_RING/ice_idc.c
Method	int ice_plug_aux_devs(struct ice_pf *pf)

```
....
691.                    ret = ice_plug_aux_dev(pf->cdev_infos[i], name);
```

Use of Zero Initialized Pointer\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=475
Status	New

The variable declared in adev at PF_RING/ice_idc.c in line 582 is not initialized when it is used by cdev_infos at PF_RING/ice_idc.c in line 670.

	Source	Destination
File	PF_RING/ice_idc.c	PF_RING/ice_idc.c
Line	625	691
Object	adev	cdev_infos

Code Snippet
File Name PF_RING/ice_idc.c
Method int ice_plug_aux_dev(struct iidc_core_dev_info *cdev_info, const char *name)

```
....
625.                cdev_info->adev = NULL;
```

File Name PF_RING/ice_idc.c
Method int ice_plug_aux_devs(struct ice_pf *pf)

```
....
691.                ret = ice_plug_aux_dev(pf->cdev_infos[i], name);
```

Use of Zero Initialized Pointer\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=476
Status	New

The variable declared in adev at PF_RING/ice_idc.c in line 644 is not initialized when it is used by cdev_infos at PF_RING/ice_idc.c in line 702.

	Source	Destination
File	PF_RING/ice_idc.c	PF_RING/ice_idc.c
Line	661	707
Object	adev	cdev_infos

Code Snippet
File Name PF_RING/ice_idc.c
Method void ice_unplug_aux_dev(struct iidc_core_dev_info *cdev_info)

```
....
661.            cdev_info->adev = NULL;
```

File Name	PF_RING/ice_idc.c
Method	void ice_unplug_aux_devs(struct ice_pf *pf)

```
....
707.                ice_unplug_aux_dev(pf->cdev_infos[i]);
```

Use of Uninitialized Pointer

Query Path:
CPP\Cx\CPP Medium Threat\Use of Uninitialized Pointer Version:0

Categories

NIST SP 800-53: SC-5 Denial of Service Protection (P1)

Description

Use of Uninitialized Pointer\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=462
Status	New

The variable declared in rule at PF_RING/ice_fdir.c in line 4134 is not initialized when it is used by rule at PF_RING/ice_fdir.c in line 4134.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4136	4141
Object	rule	rule

Code Snippet

File Name	PF_RING/ice_fdir.c
Method	ice_fdir_find_fltr_by_idx(struct ice_hw *hw, u32 fltr_idx)

```
....
4136.         struct ice_fdir_fltr *rule;
....
4141.                         return rule;
```

Use of Uninitialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=463
Status	New

The variable declared in rule at PF_RING/ice_fdir.c in line 4134 is not initialized when it is used by fltr_id at PF_RING/ice_fdir.c in line 4134.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4136	4140
Object	rule	fltr_id

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_find_fltr_by_idx(struct ice_hw *hw, u32 fltr_idx)

```
....
4136.       struct ice_fdir_fltr *rule;
....
4140.             if (fltr_idx == rule->fltr_id)
```

Use of Uninitialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=464
Status	New

The variable declared in rule at PF_RING/ice_fdir.c in line 4134 is not initialized when it is used by fltr_id at PF_RING/ice_fdir.c in line 4134.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4136	4142
Object	rule	fltr_id

Code Snippet
File Name PF_RING/ice_fdir.c
Method ice_fdir_find_fltr_by_idx(struct ice_hw *hw, u32 fltr_idx)

```
....
4136.       struct ice_fdir_fltr *rule;
....
4142.             if (fltr_idx < rule->fltr_id)
```

Use of Uninitialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=465
Status	New

The variable declared in rule at PF_RING/ice_fdir.c in line 4153 is not initialized when it is used by rule at PF_RING/ice_fdir.c in line 4153.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4155	4161
Object	rule	rule

Code Snippet
File Name PF_RING/ice_fdir.c
Method void ice_fdir_list_add_fltr(struct ice_hw *hw, struct ice_fdir_fltr *fltr)

```
....
4155.       struct ice_fdir_fltr *rule, *parent = NULL;
....
4161.               parent = rule;
```

Use of Uninitialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=466
Status	New

The variable declared in rule at PF_RING/ice_fdir.c in line 4153 is not initialized when it is used by fltr_id at PF_RING/ice_fdir.c in line 4153.

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	4155	4159
Object	rule	fltr_id

Code Snippet
File Name PF_RING/ice_fdir.c
Method void ice_fdir_list_add_fltr(struct ice_hw *hw, struct ice_fdir_fltr *fltr)

```
....
4155.       struct ice_fdir_fltr *rule, *parent = NULL;
....
4159.               if (rule->fltr_id >= fltr->fltr_id)
```

Wrong Size t Allocation

Query Path:
CPP\Cx\CPP Integer Overflow\Wrong Size t Allocation Version:0

Description

Wrong Size t Allocation\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=168
Status	New

The function prog_size in PF_RING/optimize.c at line 2939 assigns an incorrectly calculated size to a buffer, resulting in a mismatch between the value being written and the size of the buffer it is being written into.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2959	2959
Object	prog_size	prog_size

Code Snippet

File Name	PF_RING/optimize.c
Method	install_bpf_program(pcap_t *p, struct bpf_program *fp)

```
....
2959.       p->fcode.bf_insns = (struct bpf_insn *)malloc(prog_size);
```

Wrong Size t Allocation\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=169
Status	New

The function block_memsize in PF_RING/optimize.c at line 2516 assigns an incorrectly calculated size to a buffer, resulting in a mismatch between the value being written and the size of the buffer it is being written into.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2617	2617
Object	block_memsize	block_memsize

Code Snippet

File Name	PF_RING/optimize.c
Method	opt_init(opt_state_t *opt_state, struct icode *ic)

```
....
2617.       opt_state->space = (bpf_u_int32 *)malloc(block_memsize +
edge_memsize);
```

Wrong Size t Allocation\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=170
Status	New

The function edge_memsize in PF_RING/optimize.c at line 2516 assigns an incorrectly calculated size to a buffer, resulting in a mismatch between the value being written and the size of the buffer it is being written into.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2617	2617
Object	edge_memsize	edge_memsize

Code Snippet
File Name PF_RING/optimize.c
Method opt_init(opt_state_t *opt_state, struct icode *ic)

```
....
2617.        opt_state->space = (bpf_u_int32 *)malloc(block_memsize +
edge_memsize);
```

Inadequate Encryption Strength

Query Path:
CPP\Cx\CPP Medium Threat\Inadequate Encryption Strength Version:1

Categories

FISMA 2014: Configuration Management
NIST SP 800-53: SC-13 Cryptographic Protection (P1)
OWASP Top 10 2017: A3-Sensitive Data Exposure

Description

Inadequate Encryption Strength\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=478
Status	New

The application uses a weak cryptographic algorithm, MD5_Update at line 892 of PF_RING/print-tcp.c, to protect sensitive personal information ndo_sigsecret, from PF_RING/print-tcp.c at line 892.

	Source	Destination
File	PF_RING/print-tcp.c	PF_RING/print-tcp.c
Line	962	962
Object	ndo_sigsecret	MD5_Update

Code Snippet
File Name PF_RING/print-tcp.c
Method tcp_verify_signature(netdissect_options *ndo,

```
....
962.          MD5_Update(&ctx, ndo->ndo_sigsecret, strlen(ndo-
>ndo_sigsecret));
```

Inadequate Encryption Strength\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=479
Status	New

The application uses a weak cryptographic algorithm, MD5_Update at line 892 of PF_RING/print-tcp.c, to protect sensitive personal information ndo_sigsecret, from PF_RING/print-tcp.c at line 892.

	Source	Destination
File	PF_RING/print-tcp.c	PF_RING/print-tcp.c
Line	962	962
Object	ndo_sigsecret	MD5_Update

Code Snippet
File Name PF_RING/print-tcp.c
Method tcp_verify_signature(netdissect_options *ndo,

```
....
962.            MD5_Update(&ctx, ndo->ndo_sigsecret, strlen(ndo-
>ndo_sigsecret));
```

Divide By Zero

Query Path:
CPP\Cx\CPP Medium Threat\Divide By Zero Version:1
Description

Divide By Zero\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=74
Status	New

The application performs an illegal operation in i40e_update_itr, in PF_RING/i40e_txrx.c. In line 1202, the program attempts to divide by packets, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input packets in i40e_update_itr of PF_RING/i40e_txrx.c, at line 1202.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	1326	1326
Object	packets	packets

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static void i40e_update_itr(struct i40e_q_vector *q_vector,

```
....
1326.        avg_wire_size = bytes / packets;
```

Char Overflow

Query Path:
CPP\Cx\CPP Integer Overflow\Char Overflow Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.2 - Buffer overflows
NIST SP 800-53: SI-10 Information Input Validation (P1)

Description

Char Overflow\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=313
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 667 of PF_RING/pfutils.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	677	677
Object	AssignExpr	AssignExpr

Code Snippet
File Name PF_RING/pfutils.c
Method int read_packet_hex(u_char *buf, int buf_len) {

```
....
677.       c = (u_char) d;
```

Integer Overflow

Query Path:
CPP\Cx\CPP Integer Overflow\Integer Overflow Version:0

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.2 - Buffer overflows
FISMA 2014: System And Information Integrity
NIST SP 800-53: SI-10 Information Input Validation (P1)

Description

Integer Overflow\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=314

Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 458 of PF_RING/print-ntp.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	PF_RING/print-ntp.c	PF_RING/print-ntp.c
Line	468	468
Object	AssignExpr	AssignExpr

Code Snippet
File Name PF_RING/print-ntp.c
Method p_sfix(netdissect_options *ndo,

```
....
468.         f = (int)(ff * 1000000.0);    /* Treat fraction as parts per
million */
```

Double Free

Query Path:
CPP\Cx\CPP Medium Threat\Double Free Version:1

Categories

NIST SP 800-53: SI-16 Memory Protection (P1)

Description

Double Free\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=426
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2908	2898
Object	fp	fp

Code Snippet
File Name PF_RING/optimize.c
Method icode_to_fcode(struct icode *ic, struct block *root, u_int *lenp,

```
....
2908.           free(fp);
....
2898.             free(fp);
```

Wrong Memory Allocation

Query Path:
CPP\Cx\CPP Medium Threat\Wrong Memory Allocation Version:0

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)

Description

Wrong Memory Allocation\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=477
Status	New

The function malloc in PF_RING/addrtoname.c at line 708 assigns an incorrectly calculated size to a buffer, resulting in a mismatch between the value being written and the size of the buffer it is being written into.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	722	722
Object	sizeof	malloc

Code Snippet
File Name PF_RING/addrtoname.c
Method isonsap_string(netdissect_options *ndo, const uint8_t *nsap,

```
....
722.		tp->e_name = cp = (char
*)malloc(sizeof("xx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx"));
```

NULL Pointer Dereference

Query Path:
CPP\Cx\CPP Low Visibility\NULL Pointer Dereference Version:1

Categories

NIST SP 800-53: SC-5 Denial of Service Protection (P1)
OWASP Top 10 2017: A1-Injection

Description

NULL Pointer Dereference\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=171
Status	New

The variable declared in 0 at PF_RING/i40e_txrx.c in line 37 is not initialized when it is used by next_to_use at PF_RING/i40e_txrx.c in line 37.

	Source	Destination

File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	50	50
Object	0	next_to_use

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static void i40e_fdir(struct i40e_ring *tx_ring,

```
....
50.    tx_ring->next_to_use = (i < tx_ring->count) ? i : 0;
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=172
Status	New

The variable declared in 0 at PF_RING/i40e_txrx.c in line 101 is not initialized when it is used by next_to_use at PF_RING/i40e_txrx.c in line 101.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	145	145
Object	0	next_to_use

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static int i40e_program_fdir_filter(struct i40e_fdir_filter *fdir_data,

```
....
145.        tx_ring->next_to_use = ((i + 1) < tx_ring->count) ? i + 1 :
0;
```

NULL Pointer Dereference\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=173
Status	New

The variable declared in 0 at PF_RING/i40e_txrx.c in line 101 is not initialized when it is used by next_to_use at PF_RING/i40e_txrx.c in line 101.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c

Line	145	170
Object	0	next_to_use

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static int i40e_program_fdir_filter(struct i40e_fdir_filter *fdir_data,

```
....
145.        tx_ring->next_to_use = ((i + 1) < tx_ring->count) ? i + 1 :
0;
....
170.        writel(tx_ring->next_to_use, tx_ring->tail);
```

NULL Pointer Dereference\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=174
Status	New

The variable declared in 0 at PF_RING/i40e_txrx.c in line 1403 is not initialized when it is used by next_to_alloc at PF_RING/i40e_txrx.c in line 1403.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	1413	1413
Object	0	next_to_alloc

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static void i40e_reuse_rx_page(struct i40e_ring *rx_ring,

```
....
1413.       rx_ring->next_to_alloc = (nta < rx_ring->count) ? nta : 0;
```

NULL Pointer Dereference\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=175
Status	New

The variable declared in 0 at PF_RING/i40e_txrx.c in line 1436 is not initialized when it is used by next_to_alloc at PF_RING/i40e_txrx.c in line 1436.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c

Line	1446	1446
Object	0	next_to_alloc

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static void i40e_reuse_rx_skb(struct i40e_ring *rx_ring,

```
....
1446.       rx_ring->next_to_alloc = (nta < rx_ring->count) ? nta : 0;
```

NULL Pointer Dereference\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=176
Status	New

The variable declared in 0 at PF_RING/i40e_txrx.c in line 3849 is not initialized when it is used by next_to_use at PF_RING/i40e_txrx.c in line 3849.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	3864	3864
Object	0	next_to_use

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static void i40e_create_tx_ctx(struct i40e_ring *tx_ring,

```
....
3864.       tx_ring->next_to_use = (i < tx_ring->count) ? i : 0;
```

NULL Pointer Dereference\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=177
Status	New

The variable declared in 0 at PF_RING/i40e_txrx.c in line 4397 is not initialized when it is used by back at PF_RING/i40e_txrx.c in line 3360.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	4404	3410
Object	0	back

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static netdev_tx_t i40e_xmit_frame_ring(struct sk_buff *skb,

```
....
4404.       u32 tx_flags = 0;
```

▼

File Name PF_RING/i40e_txrx.c

Method static inline int i40e_tx_prepare_vlan_flags(struct sk_buff *skb,

```
....
3410.       if (!(tx_ring->vsi->back->flags & I40E_FLAG_DCB_ENABLED))
```

NULL Pointer Dereference\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=178
Status	New

The variable declared in 0 at PF_RING/i40e_txrx.c in line 4397 is not initialized when it is used by back at PF_RING/i40e_txrx.c in line 3360.

	Source	Destination
File	PF_RING/i40e_txrx.c	PF_RING/i40e_txrx.c
Line	4404	3389
Object	0	back

Code Snippet
File Name PF_RING/i40e_txrx.c
Method static netdev_tx_t i40e_xmit_frame_ring(struct sk_buff *skb,

```
....
4404.       u32 tx_flags = 0;
```

▼

File Name PF_RING/i40e_txrx.c

Method static inline int i40e_tx_prepare_vlan_flags(struct sk_buff *skb,

```
....
3389.           if (i40e_is_double_vlan(&tx_ring->vsi->back->hw) &&
```

NULL Pointer Dereference\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=179
Status	New

The variable declared in 0 at PF_RING/print-udp.c in line 215 is not initialized when it is used by rr_dv at PF_RING/print-udp.c in line 215.

	Source	Destination
File	PF_RING/print-udp.c	PF_RING/print-udp.c
Line	218	289
Object	0	rr_dv

Code Snippet

File Name	PF_RING/print-udp.c
Method	rtcp_print(netdissect_options *ndo, const u_char *hdr, const u_char *ep)

```
....
218.        const struct rtcp_rr *rr = 0;
....
289.                    GET_BE_U_4(rr->rr_dv), ts, dts);
```

NULL Pointer Dereference\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=180
Status	New

The variable declared in 0 at PF_RING/print-udp.c in line 215 is not initialized when it is used by rr_ls at PF_RING/print-udp.c in line 215.

	Source	Destination
File	PF_RING/print-udp.c	PF_RING/print-udp.c
Line	218	288
Object	0	rr_ls

Code Snippet

File Name	PF_RING/print-udp.c
Method	rtcp_print(netdissect_options *ndo, const u_char *hdr, const u_char *ep)

```
....
218.        const struct rtcp_rr *rr = 0;
....
288.                    GET_BE_U_4(rr->rr_ls),
```

NULL Pointer Dereference\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=181
Status	New

The variable declared in 0 at PF_RING/print-udp.c in line 215 is not initialized when it is used by rr_nl at PF_RING/print-udp.c in line 215.

	Source	Destination
File	PF_RING/print-udp.c	PF_RING/print-udp.c
Line	218	287
Object	0	rr_nl

Code Snippet

File Name	PF_RING/print-udp.c
Method	rtcp_print(netdissect_options *ndo, const u_char *hdr, const u_char *ep)

```
....
218.        const struct rtcp_rr *rr = 0;
....
287.                    GET_BE_U_4(rr->rr_nl) & 0x00ffffff,
```

NULL Pointer Dereference\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=182
Status	New

The variable declared in 0 at PF_RING/print-udp.c in line 215 is not initialized when it is used by rr_srcid at PF_RING/print-udp.c in line 215.

	Source	Destination
File	PF_RING/print-udp.c	PF_RING/print-udp.c
Line	218	283
Object	0	rr_srcid

Code Snippet

File Name	PF_RING/print-udp.c
Method	rtcp_print(netdissect_options *ndo, const u_char *hdr, const u_char *ep)

```
....
218.        const struct rtcp_rr *rr = 0;
....
283.                      ND_PRINT(" %u", GET_BE_U_4(rr->rr_srcid));
```

NULL Pointer Dereference\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=183
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by add_macs_to_list at PF_RING/kcompat_vfd.c in line 1069.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	1081
Object	vfd_ops	add_macs_to_list

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_mac_list_store(struct kobject *kobj,

```
....
1081.       if (!vfd_ops->add_macs_to_list || !vfd_ops-
>rem_macs_from_list)
```

NULL Pointer Dereference\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=184
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_allow_bcast at PF_RING/kcompat_vfd.c in line 2383.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	2391
Object	vfd_ops	get_allow_bcast

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.  const struct vfd_ops *vfd_ops = NULL;
```

File Name	PF_RING/kcompat_vfd.c
Method	static ssize_t vfd_allow_bcast_store(struct kobject *kobj,

```
....
2391.       if (!vfd_ops->set_allow_bcast || !vfd_ops->get_allow_bcast)
```

NULL Pointer Dereference\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=185
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_allow_bcast at PF_RING/kcompat_vfd.c in line 2346.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	2354
Object	vfd_ops	get_allow_bcast

Code Snippet

File Name	PF_RING/kcompat_vfd.c
Method	const struct vfd_ops *vfd_ops = NULL;

```
....
9.  const struct vfd_ops *vfd_ops = NULL;
```

File Name	PF_RING/kcompat_vfd.c
Method	static ssize_t vfd_allow_bcast_show(struct kobject *kobj,

```
....
2354.       if (!vfd_ops->get_allow_bcast)
```

NULL Pointer Dereference\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=186
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_allow_untagged at PF_RING/kcompat_vfd.c in line 817.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	825
Object	vfd_ops	get_allow_untagged

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.  const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c
Method static ssize_t vfd_allow_untagged_store(struct kobject *kobj,

```
....
825.        if (!vfd_ops->set_allow_untagged || !vfd_ops-
>get_allow_untagged)
```

NULL Pointer Dereference\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=187
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_allow_untagged at PF_RING/kcompat_vfd.c in line 783.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	791
Object	vfd_ops	get_allow_untagged

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.  const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_allow_untagged_show(struct kobject *kobj,

```
....
791.         if (!vfd_ops->get_allow_untagged)
```

NULL Pointer Dereference\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=188
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_egress_mirror at PF_RING/kcompat_vfd.c in line 525.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	533
Object	vfd_ops	get_egress_mirror

Code Snippet

File Name PF_RING/kcompat_vfd.c

Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_egress_mirror_store(struct kobject *kobj,

```
....
533.         if (!vfd_ops->set_egress_mirror || !vfd_ops-
>get_egress_mirror)
```

NULL Pointer Dereference\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=189
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_egress_mirror at PF_RING/kcompat_vfd.c in line 492.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c

Line	9	499
Object	vfd_ops	get_egress_mirror

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.  const struct vfd_ops *vfd_ops = NULL;
```

▼

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_egress_mirror_show(struct kobject *kobj,

```
....
499.        if (!vfd_ops->get_egress_mirror)
```

NULL Pointer Dereference\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=190
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_ingress_mirror at PF_RING/kcompat_vfd.c in line 565.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	573
Object	vfd_ops	get_ingress_mirror

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.  const struct vfd_ops *vfd_ops = NULL;
```

▼

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_ingress_mirror_show(struct kobject *kobj,

```
....
573.        if (!vfd_ops->get_ingress_mirror)
```

NULL Pointer Dereference\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=191
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_ingress_mirror at PF_RING/kcompat_vfd.c in line 599.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	607
Object	vfd_ops	get_ingress_mirror

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_ingress_mirror_store(struct kobject *kobj,

```
....
607.        if (!vfd_ops->set_ingress_mirror || !vfd_ops-
>get_ingress_mirror)
```

NULL Pointer Dereference\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=192
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_link_state at PF_RING/kcompat_vfd.c in line 1320.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	1328
Object	vfd_ops	get_link_state

Code Snippet
File Name PF_RING/kcompat_vfd.c

Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

▼

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_link_state_show(struct kobject *kobj,

```
....
1328.        if (!vfd_ops->get_link_state)
```

NULL Pointer Dereference\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=193
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_loopback at PF_RING/kcompat_vfd.c in line 852.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	859
Object	vfd_ops	get_loopback

Code Snippet

File Name PF_RING/kcompat_vfd.c

Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

▼

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_loopback_show(struct kobject *kobj,

```
....
859.         if (!vfd_ops->get_loopback)
```

NULL Pointer Dereference\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=194
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_loopback at PF_RING/kcompat_vfd.c in line 885.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	893
Object	vfd_ops	get_loopback

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c
Method static ssize_t vfd_loopback_store(struct kobject *kobj,

```
....
893.         if (!vfd_ops->set_loopback || !vfd_ops->get_loopback)
```

NULL Pointer Dereference\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=195
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_mac at PF_RING/kcompat_vfd.c in line 920.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	927
Object	vfd_ops	get_mac

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_mac_show(struct kobject *kobj, struct kobj_attribute *attr,

```
....
927.        if (!vfd_ops->get_mac)
```

NULL Pointer Dereference\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=196
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_mac at PF_RING/kcompat_vfd.c in line 950.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	959
Object	vfd_ops	get_mac

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.  const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_mac_store(struct kobject *kobj,

```
....
959.        if (!vfd_ops->set_mac || !vfd_ops->get_mac)
```

NULL Pointer Dereference\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=197
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_mac_anti_spoof at PF_RING/kcompat_vfd.c in line 676.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c

Line	9	684
Object	vfd_ops	get_mac_anti_spoof

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.  const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c
Method static ssize_t vfd_mac_anti_spoof_store(struct kobject *kobj,

```
....
684.         if (!vfd_ops->set_mac_anti_spoof || !vfd_ops-
>get_mac_anti_spoof)
```

NULL Pointer Dereference\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=198
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_mac_anti_spoof at PF_RING/kcompat_vfd.c in line 639.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	647
Object	vfd_ops	get_mac_anti_spoof

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.  const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c
Method static ssize_t vfd_mac_anti_spoof_show(struct kobject *kobj,

```
....
647.         if (!vfd_ops->get_mac_anti_spoof)
```

NULL Pointer Dereference\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=199
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_mac_list at PF_RING/kcompat_vfd.c in line 992.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	1004
Object	vfd_ops	get_mac_list

Code Snippet

File Name PF_RING/kcompat_vfd.c

Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_mac_list_show(struct kobject *kobj,

```
....
1004.        if (!vfd_ops->get_mac_list)
```

NULL Pointer Dereference\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=200
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_max_tx_rate at PF_RING/kcompat_vfd.c in line 1499.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	1506
Object	vfd_ops	get_max_tx_rate

Code Snippet

File Name PF_RING/kcompat_vfd.c

Method	const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name	PF_RING/kcompat_vfd.c
Method	static ssize_t vfd_max_tx_rate_show(struct kobject *kobj,

```
....
1506.        if (!vfd_ops->get_max_tx_rate)
```

NULL Pointer Dereference\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=201
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_min_tx_rate at PF_RING/kcompat_vfd.c in line 1561.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	1564
Object	vfd_ops	get_min_tx_rate

Code Snippet

File Name	PF_RING/kcompat_vfd.c
Method	const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name	PF_RING/kcompat_vfd.c
Method	static ssize_t vfd_min_tx_rate_show(struct kobject *kobj,

```
....
1564.        if (!vfd_ops->get_min_tx_rate)
```

NULL Pointer Dereference\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=202
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_num_queues at PF_RING/kcompat_vfd.c in line 2230.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	2237
Object	vfd_ops	get_num_queues

Code Snippet

File Name PF_RING/kcompat_vfd.c

Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_num_queues_show(struct kobject *kobj,

```
....
2237.        if (!vfd_ops->get_num_queues)
```

NULL Pointer Dereference\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=203
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_pf_egress_mirror at PF_RING/kcompat_vfd.c in line 2132.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	2140
Object	vfd_ops	get_pf_egress_mirror

Code Snippet

File Name PF_RING/kcompat_vfd.c

Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method	static ssize_t pf_egress_mirror_store(struct kobject *kobj,

```
....
2140.        if (!vfd_ops->set_pf_egress_mirror || !vfd_ops-
>get_pf_egress_mirror)
```

NULL Pointer Dereference\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=204
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_pf_egress_mirror at PF_RING/kcompat_vfd.c in line 2100.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	2106
Object	vfd_ops	get_pf_egress_mirror

Code Snippet

File Name	PF_RING/kcompat_vfd.c
Method	const struct vfd_ops *vfd_ops = NULL;

```
....
9.  const struct vfd_ops *vfd_ops = NULL;
```

File Name	PF_RING/kcompat_vfd.c
Method	static ssize_t pf_egress_mirror_show(struct kobject *kobj,

```
....
2106.        if (!vfd_ops->get_pf_egress_mirror)
```

NULL Pointer Dereference\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=205
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_pf_ingress_mirror at PF_RING/kcompat_vfd.c in line 2064.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c

Line	9	2072
Object	vfd_ops	get_pf_ingress_mirror

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c
Method static ssize_t pf_ingress_mirror_store(struct kobject *kobj,

```
....
2072.        if (!vfd_ops->set_pf_ingress_mirror || !vfd_ops-
>get_pf_ingress_mirror)
```

NULL Pointer Dereference\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=206
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_pf_ingress_mirror at PF_RING/kcompat_vfd.c in line 2032.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	2038
Object	vfd_ops	get_pf_ingress_mirror

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c
Method static ssize_t pf_ingress_mirror_show(struct kobject *kobj,

```
....
2038.        if (!vfd_ops->get_pf_ingress_mirror)
```

NULL Pointer Dereference\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=207
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_pf_qos_tc_lsp at PF_RING/kcompat_vfd.c in line 2568.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	2577
Object	vfd_ops	get_pf_qos_tc_lsp

Code Snippet

File Name PF_RING/kcompat_vfd.c

Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t pf_qos_tc_lsp_store(struct kobject *kobj,

```
....
2577.        if (!vfd_ops->set_pf_qos_tc_lsp || !vfd_ops-
>get_pf_qos_tc_lsp)
```

NULL Pointer Dereference\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=208
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_pf_qos_tc_lsp at PF_RING/kcompat_vfd.c in line 2534.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	2542
Object	vfd_ops	get_pf_qos_tc_lsp

Code Snippet

File Name	PF_RING/kcompat_vfd.c
Method	const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

▼

File Name	PF_RING/kcompat_vfd.c
Method	static ssize_t pf_qos_tc_lsp_show(struct kobject *kobj,

```
....
2542.        if (!vfd_ops->set_pf_qos_tc_lsp || !vfd_ops-
>get_pf_qos_tc_lsp)
```

NULL Pointer Dereference\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=209
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_pf_qos_tc_max_bw at PF_RING/kcompat_vfd.c in line 2601.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	2608
Object	vfd_ops	get_pf_qos_tc_max_bw

Code Snippet

File Name	PF_RING/kcompat_vfd.c
Method	const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

▼

File Name	PF_RING/kcompat_vfd.c
Method	static ssize_t pf_qos_tc_max_bw_show(struct kobject *kobj,

```
....
2608.        if (!vfd_ops->set_pf_qos_tc_max_bw || !vfd_ops-
>get_pf_qos_tc_max_bw)
```

NULL Pointer Dereference\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=210
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_pf_qos_tc_max_bw at PF_RING/kcompat_vfd.c in line 2634.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	2642
Object	vfd_ops	get_pf_qos_tc_max_bw

Code Snippet

File Name	PF_RING/kcompat_vfd.c
Method	const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name	PF_RING/kcompat_vfd.c
Method	static ssize_t pf_qos_tc_max_bw_store(struct kobject *kobj,

```
....
2642.        if (!vfd_ops->set_pf_qos_tc_max_bw || !vfd_ops-
>get_pf_qos_tc_max_bw)
```

NULL Pointer Dereference\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=211
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_pf_qos_tc_priority at PF_RING/kcompat_vfd.c in line 2479.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	2490
Object	vfd_ops	get_pf_qos_tc_priority

Code Snippet

File Name	PF_RING/kcompat_vfd.c
Method	const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

▼

File Name PF_RING/kcompat_vfd.c

Method static ssize_t pf_qos_tc_priority_store(struct kobject *kobj,

```
....
2490.            !vfd_ops->get_pf_qos_tc_priority)
```

NULL Pointer Dereference\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=212
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_pf_qos_tc_priority at PF_RING/kcompat_vfd.c in line 2435.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	2445
Object	vfd_ops	get_pf_qos_tc_priority

Code Snippet

File Name PF_RING/kcompat_vfd.c

Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

▼

File Name PF_RING/kcompat_vfd.c

Method static ssize_t pf_qos_tc_priority_show(struct kobject *kobj,

```
....
2445.            !vfd_ops->get_pf_qos_tc_priority)
```

NULL Pointer Dereference\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=213
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_pf_tpid at PF_RING/kcompat_vfd.c in line 2168.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	2175
Object	vfd_ops	get_pf_tpid

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t pf_tpid_show(struct kobject *kobj,

```
....
2175.        if (!vfd_ops->get_pf_tpid)
```

NULL Pointer Dereference\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=214
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_promisc at PF_RING/kcompat_vfd.c in line 1156.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	1163
Object	vfd_ops	get_promisc

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_promisc_show(struct kobject *kobj,

```
....
1163.        if (!vfd_ops->get_promisc)
```

NULL Pointer Dereference\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=215
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_promisc at PF_RING/kcompat_vfd.c in line 1193.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	1203
Object	vfd_ops	get_promisc

Code Snippet

File Name PF_RING/kcompat_vfd.c

Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

▼

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_promisc_store(struct kobject *kobj,

```
....
1203.        if (!vfd_ops->get_promisc || !vfd_ops->set_promisc)
```

NULL Pointer Dereference\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=216
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_queue_type at PF_RING/kcompat_vfd.c in line 2302.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c

Line	9	2309
Object	vfd_ops	get_queue_type

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.  const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_queue_type_show(struct kobject *kobj,

```
....
2309.       if (!vfd_ops->get_queue_type)
```

NULL Pointer Dereference\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=217
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_rx_bytes at PF_RING/kcompat_vfd.c in line 1695.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	1702
Object	vfd_ops	get_rx_bytes

Code Snippet
File Name PF_RING/kcompat_vfd.c
Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.  const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_rx_bytes_show(struct kobject *kobj,

```
....
1702.       if (!vfd_ops->get_rx_bytes)
```

NULL Pointer Dereference\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=218
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_rx_dropped at PF_RING/kcompat_vfd.c in line 1724.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	1731
Object	vfd_ops	get_rx_dropped

Code Snippet

File Name	PF_RING/kcompat_vfd.c
Method	const struct vfd_ops *vfd_ops = NULL;

```
....
9.  const struct vfd_ops *vfd_ops = NULL;
```

File Name	PF_RING/kcompat_vfd.c
Method	static ssize_t vfd_rx_dropped_show(struct kobject *kobj,

```
....
1731.       if (!vfd_ops->get_rx_dropped)
```

NULL Pointer Dereference\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=219
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_rx_packets at PF_RING/kcompat_vfd.c in line 1753.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	1760
Object	vfd_ops	get_rx_packets

Code Snippet

File Name	PF_RING/kcompat_vfd.c
Method	const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_rx_packets_show(struct kobject *kobj,

```
....
1760.        if (!vfd_ops->get_rx_packets)
```

NULL Pointer Dereference\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=220
Status	New

The variable declared in vfd_ops at PF_RING/kcompat_vfd.c in line 9 is not initialized when it is used by get_trunk at PF_RING/kcompat_vfd.c in line 319.

	Source	Destination
File	PF_RING/kcompat_vfd.c	PF_RING/kcompat_vfd.c
Line	9	328
Object	vfd_ops	get_trunk

Code Snippet

File Name PF_RING/kcompat_vfd.c

Method const struct vfd_ops *vfd_ops = NULL;

```
....
9.   const struct vfd_ops *vfd_ops = NULL;
```

File Name PF_RING/kcompat_vfd.c

Method static ssize_t vfd_trunk_show(struct kobject *kobj,

```
....
328.        if (!vfd_ops->get_trunk)
```

Unchecked Return Value

Query Path:

CPP\Cx\CPP Low Visibility\Unchecked Return Value Version:1

Categories

NIST SP 800-53: SI-11 Error Handling (P2)

Description

Unchecked Return Value\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=4
Status	New

The ieee8021q_tci_string method calls the snprintf function, at line 1325 of PF_RING/addrtoname.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	1328	1328
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/addrtoname.c
Method	ieee8021q_tci_string(const uint16_t tci)

```
....
1328.        snprintf(buf, sizeof(buf), "vlan %u, p %u%s",
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=5
Status	New

The etheraddr_string method calls the snprintf function, at line 591 of PF_RING/addrtoname.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	631	631
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/addrtoname.c
Method	etheraddr_string(netdissect_options *ndo, const uint8_t *ep)

```
....
631.                 snprintf(cp, BUFSIZE - (2 + 5*3), " (oui %s)",
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=6
Status	New

The tcpport_string method calls the snprintf function, at line 739 of PF_RING/addrtoname.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	752	752
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/addrtoname.c
Method	tcpport_string(netdissect_options *ndo, u_short port)

```
....
752.          (void)snprintf(buf, sizeof(buf), "%u", i);
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=7
Status	New

The udpport_string method calls the snprintf function, at line 761 of PF_RING/addrtoname.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	774	774
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/addrtoname.c
Method	udpport_string(netdissect_options *ndo, u_short port)

```
....
774.          (void)snprintf(buf, sizeof(buf), "%u", i);
```

Unchecked Return Value\Path 5:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=8
Status	New

The init_servarray method calls the snprintf function, at line 812 of PF_RING/addrtoname.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	832	832
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/addrtoname.c
Method	init_servarray(netdissect_options *ndo)

```
....
832.                    (void)snprintf(buf, sizeof(buf), "%d", port);
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=9
Status	New

The icode_to_fcode method calls the snprintf function, at line 2872 of PF_RING/optimize.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2896	2896
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/optimize.c
Method	icode_to_fcode(struct icode *ic, struct block *root, u_int *lenp,

```
....
2896.               (void)snprintf(errbuf, PCAP_ERRBUF_SIZE,
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=10
Status	New

The install_bpf_program method calls the snprintf function, at line 2939 of PF_RING/optimize.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2947	2947
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/optimize.c
Method	install_bpf_program(pcap_t *p, struct bpf_program *fp)

```
....
2947.                   snprintf(p->errbuf, sizeof(p->errbuf),
```

Unchecked Return Value\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=11
Status	New

The Parse_fh method calls the snprintf function, at line 85 of PF_RING/parsenfsfh.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/parsenfsfh.c	PF_RING/parsenfsfh.c
Line	405	405
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/parsenfsfh.c
Method	Parse_fh(netdissect_options *ndo, const unsigned char *fh, u_int len,

```
....
405.                   (void)snprintf(&(fsidp->Opaque_Handle[i*2]), 3,
"%.2X",
```

Unchecked Return Value\Path 9:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=12
Status	New

The *msec2dhmsm method calls the snprintf function, at line 439 of PF_RING/pfutils.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	440	440
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/pfutils.c
Method char *msec2dhmsm(u_int64_t msec, char *buf, u_int buf_len) {

```
....
440.    snprintf(buf, buf_len, "%u:%02u:%02u:%02u:%03u",
```

Unchecked Return Value\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=13
Status	New

The busid2node method calls the snprintf function, at line 482 of PF_RING/pfutils.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	490	490
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/pfutils.c
Method int busid2node(int slot, int bus, int device, int function) {

```
....
490.    snprintf(path, sizeof(path),
"/sys/bus/pci/devices/%04X:%02X:%02X.%X/numa_node",
```

Unchecked Return Value\Path 11:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=14
Status	New

The trace method calls the snprintf function, at line 585 of PF_RING/pfutils.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	612	612
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/pfutils.c
Method void trace(int trace_level, char *file, int line, char * format, ...) {

```
....
612.     snprintf(out_buf, sizeof(out_buf), "%s [%s:%d] %s%s", theDate,
file, line, extra_msg, buf);
```

Unchecked Return Value\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=15
Status	New

The format_id method calls the snprintf function, at line 128 of PF_RING/print-babel.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-babel.c	PF_RING/print-babel.c
Line	131	131
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-babel.c
Method format_id(netdissect_options *ndo, const u_char *id)

```
....
131.       snprintf(buf, 25, "%02x:%02x:%02x:%02x:%02x:%02x:%02x:%02x",
```

Unchecked Return Value\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=16
Status	New

The format_prefix method calls the snprintf function, at line 143 of PF_RING/print-babel.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-babel.c	PF_RING/print-babel.c
Line	153	153
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-babel.c
Method format_prefix(netdissect_options *ndo, const u_char *prefix, unsigned char plen)

```
....
153.          snprintf(buf, 50, "%s/%u", ipaddr_string(ndo, prefix + 
12), plen - 96);
```

Unchecked Return Value\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=17
Status	New

The format_prefix method calls the snprintf function, at line 143 of PF_RING/print-babel.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-babel.c	PF_RING/print-babel.c
Line	155	155
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-babel.c
Method format_prefix(netdissect_options *ndo, const u_char *prefix, unsigned char plen)

```
....
155.          snprintf(buf, 50, "%s/%u", ip6addr_string(ndo, prefix), 
plen);
```

Unchecked Return Value\Path 15:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=18
Status	New

The format_interval method calls the snprintf function, at line 175 of PF_RING/print-babel.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-babel.c	PF_RING/print-babel.c
Line	181	181
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-babel.c
Method format_interval(const uint16_t i)

```
....
181.        snprintf(buf, sizeof(buf), "%u.%02us", i / 100, i % 100);
```

Unchecked Return Value\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=19
Status	New

The format_timestamp method calls the snprintf function, at line 192 of PF_RING/print-babel.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-babel.c	PF_RING/print-babel.c
Line	195	195
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-babel.c
Method format_timestamp(const uint32_t i)

```
....
195.        snprintf(buf, sizeof(buf), "%u.%06us", i / 1000000, i %
1000000);
```

Unchecked Return Value\Path 17:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=20
Status	New

The as_printf method calls the snprintf function, at line 572 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	576	576
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method as_printf(netdissect_options *ndo,

```
....
576.            snprintf(str, size, "%u", asnum);
```

Unchecked Return Value\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=21
Status	New

The as_printf method calls the snprintf function, at line 572 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	578	578
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method as_printf(netdissect_options *ndo,

```
....
578.            snprintf(str, size, "%u.%u", asnum >> 16, asnum & 0xFFFF);
```

Unchecked Return Value\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=22
Status	New

The decode_prefix4 method calls the snprintf function, at line 586 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	605	605
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method decode_prefix4(netdissect_options *ndo,

```
....
605.       snprintf(buf, buflen, "%s/%u", ipaddr_string(ndo, (const
u_char *)&addr), plen);
```

Unchecked Return Value\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=23
Status	New

The decode_labeled_prefix4 method calls the snprintf function, at line 613 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	650	650
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method decode_labeled_prefix4(netdissect_options *ndo,

```
....
650.       snprintf(buf, buflen, "%s/%u, label:%u %s",
```

Unchecked Return Value\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=24
Status	New

The bgp_vpn_ip_print method calls the snprintf function, at line 671 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	681	681
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method bgp_vpn_ip_print(netdissect_options *ndo,

```
....
681.            snprintf(pos, sizeof(addr), "%s",
GET_IPADDR_STRING(pptr));
```

Unchecked Return Value\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=25
Status	New

The bgp_vpn_ip_print method calls the snprintf function, at line 671 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	684	684
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method bgp_vpn_ip_print(netdissect_options *ndo,

```
....
684.            snprintf(pos, sizeof(addr), "%s",
GET_IP6ADDR_STRING(pptr));
```

Unchecked Return Value\Path 23:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=26
Status	New

The bgp_vpn_ip_print method calls the snprintf function, at line 671 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	687	687
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method bgp_vpn_ip_print(netdissect_options *ndo,

```
....
687.            snprintf(pos, sizeof(addr), "bogus address length %u",
addr_length);
```

Unchecked Return Value\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=27
Status	New

The bgp_vpn_sg_print method calls the snprintf function, at line 715 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	732	732
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method bgp_vpn_sg_print(netdissect_options *ndo,

```
....
732.            snprintf(buf + offset, buflen - offset, ", Source %s",
```

Unchecked Return Value\Path 25:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=28
Status	New

The bgp_vpn_sg_print method calls the snprintf function, at line 715 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	746	746
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method bgp_vpn_sg_print(netdissect_options *ndo,

```
....
746.            snprintf(buf + offset, buflen - offset, ", Group %s",
```

Unchecked Return Value\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=29
Status	New

The bgp_vpn_rd_print method calls the snprintf function, at line 757 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	771	771
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method bgp_vpn_rd_print(netdissect_options *ndo,

```
....
771.            snprintf(pos, sizeof(rd) - (pos - rd), "%u:%u (=
%u.%u.%u.%u)",
```

Unchecked Return Value\Path 27:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=30
Status	New

The bgp_vpn_rd_print method calls the snprintf function, at line 757 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	780	780
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/print-bgp.c
Method	bgp_vpn_rd_print(netdissect_options *ndo,

```
....
780.            snprintf(pos, sizeof(rd) - (pos - rd), "%u.%u.%u.%u:%u",
```

Unchecked Return Value\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=31
Status	New

The bgp_vpn_rd_print method calls the snprintf function, at line 757 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	788	788
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/print-bgp.c
Method	bgp_vpn_rd_print(netdissect_options *ndo,

```
....
788.            snprintf(pos, sizeof(rd) - (pos - rd), "%s:%u
(%u.%u.%u.%u:%u)",
```

Unchecked Return Value\Path 29:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=32
Status	New

The bgp_vpn_rd_print method calls the snprintf function, at line 757 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	795	795
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method bgp_vpn_rd_print(netdissect_options *ndo,

```
....
795.          snprintf(pos, sizeof(rd) - (pos - rd), "unknown RD
format");
```

Unchecked Return Value\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=33
Status	New

The bgp_rt_prefix_print method calls the snprintf function, at line 913 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	928	928
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method bgp_rt_prefix_print(netdissect_options *ndo,

```
....
928.          snprintf(output, sizeof(output), "route-target: 0:0/0");
```

Unchecked Return Value\Path 31:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=34
Status	New

The bgp_rt_prefix_print method calls the snprintf function, at line 913 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	946	946
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/print-bgp.c
Method	bgp_rt_prefix_print(netdissect_options *ndo,

```
....
946.          snprintf(output, sizeof(output), "route-target: partial-
type: (%s/%d)",
```

Unchecked Return Value\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=35
Status	New

The bgp_rt_prefix_print method calls the snprintf function, at line 913 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	960	960
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/print-bgp.c
Method	bgp_rt_prefix_print(netdissect_options *ndo,

```
....
960.          snprintf(output, sizeof(output), "route-target: %u:%u/%d
(%s)",
```

Unchecked Return Value\Path 33:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=36
Status	New

The bgp_rt_prefix_print method calls the snprintf function, at line 913 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	968	968
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/print-bgp.c
Method	bgp_rt_prefix_print(netdissect_options *ndo,

```
....
968.            snprintf(output, sizeof(output), "route-target:
%u.%u.%u.%u:%u/%d (%s)",
```

Unchecked Return Value\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=37
Status	New

The bgp_rt_prefix_print method calls the snprintf function, at line 913 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	975	975
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/print-bgp.c
Method	bgp_rt_prefix_print(netdissect_options *ndo,

```
....
975.            snprintf(output, sizeof(output), "route-target: %s:%u/%d
(%s)",
```

Unchecked Return Value\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=38
Status	New

The bgp_rt_prefix_print method calls the snprintf function, at line 913 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	981	981
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method bgp_rt_prefix_print(netdissect_options *ndo,

```
....
981.          snprintf(output, sizeof(output), "route target: unknown-
type(%04x) (%s/%d)",
```

Unchecked Return Value\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=39
Status	New

The decode_labeled_vpn_prefix4 method calls the snprintf function, at line 1048 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1071	1071
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method decode_labeled_vpn_prefix4(netdissect_options *ndo,

```
....
1071.       snprintf(buf, buflen, "RD: %s, %s/%u, label:%u %s",
```

Unchecked Return Value\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=40
Status	New

The decode_mdt_vpn_nlri method calls the snprintf function, at line 1094 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1115	1115
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method decode_mdt_vpn_nlri(netdissect_options *ndo,

```
....
1115.       snprintf(buf, buflen, "RD: %s, VPN IP Address: %s, MC Group
Address: %s",
```

Unchecked Return Value\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=41
Status	New

The decode_multicast_vpn method calls the snprintf function, at line 1144 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1158	1158
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method decode_multicast_vpn(netdissect_options *ndo,

```
....
1158.       snprintf(buf, buflen, "Route-Type: %s (%u), length: %u",
```

Unchecked Return Value\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=42
Status	New

The decode_multicast_vpn method calls the snprintf function, at line 1144 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1167	1167
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/print-bgp.c
Method	decode_multicast_vpn(netdissect_options *ndo,

```
....
1167.           snprintf(buf + offset, buflen - offset, ", RD: %s,
Originator %s",
```

Unchecked Return Value\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=43
Status	New

The decode_multicast_vpn method calls the snprintf function, at line 1144 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1175	1175
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/print-bgp.c
Method	decode_multicast_vpn(netdissect_options *ndo,

```
....
1175.           snprintf(buf + offset, buflen - offset, ", RD: %s,
Source-AS %s",
```

Unchecked Return Value\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=44
Status	New

The decode_multicast_vpn method calls the snprintf function, at line 1144 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1184	1184
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method decode_multicast_vpn(netdissect_options *ndo,

```
....
1184.          snprintf(buf + offset, buflen - offset, ", RD: %s",
```

Unchecked Return Value\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=45
Status	New

The decode_multicast_vpn method calls the snprintf function, at line 1144 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1193	1193
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method decode_multicast_vpn(netdissect_options *ndo,

```
....
1193.          snprintf(buf + offset, buflen - offset, ", Originator
%s",
```

Unchecked Return Value\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=46
Status	New

The decode_multicast_vpn method calls the snprintf function, at line 1144 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1200	1200
Object	snprintf	snprintf

Code Snippet

File Name PF_RING/print-bgp.c

Method decode_multicast_vpn(netdissect_options *ndo,

```
....
1200.          snprintf(buf + offset, buflen - offset, ", RD: %s",
```

Unchecked Return Value\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=47
Status	New

The decode_multicast_vpn method calls the snprintf function, at line 1144 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1211	1211
Object	snprintf	snprintf

Code Snippet

File Name PF_RING/print-bgp.c

Method decode_multicast_vpn(netdissect_options *ndo,

```
....
1211.          snprintf(buf + offset, buflen - offset, ", RD: %s,
Source-AS %s",
```

Unchecked Return Value\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=48
Status	New

The decode_prefix6 method calls the snprintf function, at line 1366 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1386	1386
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/print-bgp.c
Method	decode_prefix6(netdissect_options *ndo,

```
....
1386.       snprintf(buf, buflen, "%s/%u", ip6addr_string(ndo, (const
u_char *)&addr), plen);
```

Unchecked Return Value\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=49
Status	New

The decode_labeled_prefix6 method calls the snprintf function, at line 1394 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1422	1422
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/print-bgp.c
Method	decode_labeled_prefix6(netdissect_options *ndo,

```
....
1422.       snprintf(buf, buflen, "%s/%u, label:%u %s",
```

Unchecked Return Value\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=50
Status	New

The decode_labeled_vpn_prefix6 method calls the snprintf function, at line 1438 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1461	1461
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method decode_labeled_vpn_prefix6(netdissect_options *ndo,

```
....
1461.      snprintf(buf, buflen, "RD: %s, %s/%u, label:%u %s",
```

Unchecked Return Value\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=51
Status	New

The decode_clnp_prefix method calls the snprintf function, at line 1472 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1490	1490
Object	snprintf	snprintf

Code Snippet
File Name PF_RING/print-bgp.c
Method decode_clnp_prefix(netdissect_options *ndo,

```
....
1490.      snprintf(buf, buflen, "%s/%u",
```

Unchecked Return Value\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=52
Status	New

The decode_labeled_vpn_clnp_prefix method calls the snprintf function, at line 1498 of PF_RING/print-bgp.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-bgp.c	PF_RING/print-bgp.c
Line	1521	1521
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/print-bgp.c
Method	decode_labeled_vpn_clnp_prefix(netdissect_options *ndo,

```
....
1521.        snprintf(buf, buflen, "RD: %s, %s/%u, label:%u %s",
```

Unchecked Return Value\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=53
Status	New

The ns_rcode method calls the snprintf function, at line 55 of PF_RING/print-domain.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	PF_RING/print-domain.c	PF_RING/print-domain.c
Line	61	61
Object	snprintf	snprintf

Code Snippet

File Name	PF_RING/print-domain.c
Method	ns_rcode(u_int rcode) {

```
....
61.   snprintf(buf, sizeof(buf), " Resp%u", rcode & 0xfff);
```

Use of Sizeof On a Pointer Type

Query Path:

CPP\Cx\CPP Low Visibility\Use of Sizeof On a Pointer Type Version:1

Description

Use of Sizeof On a Pointer Type\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=516
Status	New

	Source	Destination
File	PF_RING/i40e_client.c	PF_RING/i40e_client.c
Line	297	319
Object	platform_data	sizeof

Code Snippet

File Name PF_RING/i40e_client.c

Method static int i40e_init_peer_mfd_devices(struct i40e_pf *pf)

```
....
297.         struct i40e_peer_dev_platform_data *platform_data;
....
319.                 i40e_mfd_cells[i].pdata_size = sizeof(platform_data);
```

Use of Sizeof On a Pointer Type\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=517
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	291
Object	eigrp_tlv_header	sizeof

Code Snippet

File Name PF_RING/print-eigrp.c

Method eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.      const struct eigrp_tlv_header *eigrp_tlv_header;
....
291.           if (eigrp_tlv_len < sizeof(struct eigrp_tlv_header) ||
```

Use of Sizeof On a Pointer Type\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=518
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	284
Object	eigrp_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-eigrp.c
Method eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.      const struct eigrp_tlv_header *eigrp_tlv_header;
....
284.          ND_TCHECK_LEN(tptr, sizeof(struct eigrp_tlv_header));
```

Use of Sizeof On a Pointer Type\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=519
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	293
Object	eigrp_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-eigrp.c
Method eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.      const struct eigrp_tlv_header *eigrp_tlv_header;
....
293.              print_unknown_data(ndo,tptr+sizeof(struct
eigrp_tlv_header),"\n\t    ",tlen);
```

Use of Sizeof On a Pointer Type\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=520
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	304
Object	eigrp_tlv_header	sizeof

Code Snippet

File Name	PF_RING/print-eigrp.c
Method	eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.      const struct eigrp_tlv_header *eigrp_tlv_header;
....
304.          if (eigrp_tlv_len < sizeof(struct eigrp_tlv_header)) {
```

Use of Sizeof On a Pointer Type\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=521
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	306
Object	eigrp_tlv_header	sizeof

Code Snippet

File Name	PF_RING/print-eigrp.c
Method	eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.      const struct eigrp_tlv_header *eigrp_tlv_header;
....
306.                           sizeof(struct eigrp_tlv_header));
```

Use of Sizeof On a Pointer Type\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=522
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	309

Object	eigrp_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-eigrp.c
Method eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.       const struct eigrp_tlv_header *eigrp_tlv_header;
....
309.           tlv_tptr=tptr+sizeof(struct eigrp_tlv_header);
```

Use of Sizeof On a Pointer Type\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=523
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	310
Object	eigrp_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-eigrp.c
Method eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.       const struct eigrp_tlv_header *eigrp_tlv_header;
....
310.           tlv_tlen=eigrp_tlv_len-sizeof(struct eigrp_tlv_header);
```

Use of Sizeof On a Pointer Type\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=524
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	321
Object	eigrp_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-eigrp.c
Method eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.        const struct eigrp_tlv_header *eigrp_tlv_header;
....
321.                          sizeof(struct eigrp_tlv_header) +
sizeof(*tlv_ptr.eigrp_tlv_general_parm));
```

Use of Sizeof On a Pointer Type\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=525
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	338
Object	eigrp_tlv_header	sizeof

Code Snippet

File Name	PF_RING/print-eigrp.c
Method	eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.        const struct eigrp_tlv_header *eigrp_tlv_header;
....
338.                                 sizeof(struct eigrp_tlv_header) +
sizeof(*tlv_ptr.eigrp_tlv_sw_version));
```

Use of Sizeof On a Pointer Type\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=526
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	353
Object	eigrp_tlv_header	sizeof

Code Snippet

File Name	PF_RING/print-eigrp.c
Method	eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.       const struct eigrp_tlv_header *eigrp_tlv_header;
....
353.                               sizeof(struct eigrp_tlv_header) +
sizeof(*tlv_ptr.eigrp_tlv_ip_int));
```

Use of Sizeof On a Pointer Type\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=527
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	388
Object	eigrp_tlv_header	sizeof

Code Snippet

File Name	PF_RING/print-eigrp.c
Method	eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.       const struct eigrp_tlv_header *eigrp_tlv_header;
....
388.                               sizeof(struct eigrp_tlv_header) +
sizeof(*tlv_ptr.eigrp_tlv_ip_ext));
```

Use of Sizeof On a Pointer Type\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=528
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	431
Object	eigrp_tlv_header	sizeof

Code Snippet

File Name	PF_RING/print-eigrp.c
Method	eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.        const struct eigrp_tlv_header *eigrp_tlv_header;
....
431.                                  sizeof(struct eigrp_tlv_header) +
sizeof(*tlv_ptr.eigrp_tlv_at_cable_setup));
```

Use of Sizeof On a Pointer Type\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=529
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	445
Object	eigrp_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-eigrp.c
Method eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.        const struct eigrp_tlv_header *eigrp_tlv_header;
....
445.                                  sizeof(struct eigrp_tlv_header) +
sizeof(*tlv_ptr.eigrp_tlv_at_int));
```

Use of Sizeof On a Pointer Type\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=530
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	473
Object	eigrp_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-eigrp.c
Method eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.        const struct eigrp_tlv_header *eigrp_tlv_header;
....
473.                                sizeof(struct eigrp_tlv_header) +
sizeof(*tlv_ptr.eigrp_tlv_at_ext));
```

Use of Sizeof On a Pointer Type\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=531
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	523
Object	eigrp_tlv_header	sizeof

Code Snippet

File Name	PF_RING/print-eigrp.c
Method	eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.        const struct eigrp_tlv_header *eigrp_tlv_header;
....
523.                print_unknown_data(ndo,tptr+sizeof(struct
eigrp_tlv_header),"\n\t    ",
```

Use of Sizeof On a Pointer Type\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=532
Status	New

	Source	Destination
File	PF_RING/print-eigrp.c	PF_RING/print-eigrp.c
Line	218	524
Object	eigrp_tlv_header	sizeof

Code Snippet

File Name	PF_RING/print-eigrp.c
Method	eigrp_print(netdissect_options *ndo, const u_char *pptr, u_int len)

```
....
218.        const struct eigrp_tlv_header *eigrp_tlv_header;
....
524.                                        eigrp_tlv_len-sizeof(struct
eigrp_tlv_header));
```

Use of Sizeof On a Pointer Type\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=533
Status	New

	Source	Destination
File	PF_RING/print-ldp.c	PF_RING/print-ldp.c
Line	573	626
Object	ldp_msg_header	sizeof

Code Snippet

File Name	PF_RING/print-ldp.c
Method	ldp_pdu_print(netdissect_options *ndo,

```
....
573.        const struct ldp_msg_header *ldp_msg_header;
....
626.            if (msg_len < sizeof(struct ldp_msg_header)-4) {
```

Use of Sizeof On a Pointer Type\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=534
Status	New

	Source	Destination
File	PF_RING/print-ldp.c	PF_RING/print-ldp.c
Line	573	620
Object	ldp_msg_header	sizeof

Code Snippet

File Name	PF_RING/print-ldp.c
Method	ldp_pdu_print(netdissect_options *ndo,

```
....
573.        const struct ldp_msg_header *ldp_msg_header;
....
620.            ND_TCHECK_LEN(tptr, sizeof(struct ldp_msg_header));
```

Use of Sizeof On a Pointer Type\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=535
Status	New

	Source	Destination
File	PF_RING/print-ldp.c	PF_RING/print-ldp.c
Line	573	635
Object	ldp_msg_header	sizeof

Code Snippet
File Name PF_RING/print-ldp.c
Method ldp_pdu_print(netdissect_options *ndo,

```
....
573.        const struct ldp_msg_header *ldp_msg_header;
....
635.                         sizeof(struct ldp_msg_header)-4);
```

Use of Sizeof On a Pointer Type\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=536
Status	New

	Source	Destination
File	PF_RING/print-ldp.c	PF_RING/print-ldp.c
Line	573	649
Object	ldp_msg_header	sizeof

Code Snippet
File Name PF_RING/print-ldp.c
Method ldp_pdu_print(netdissect_options *ndo,

```
....
573.        const struct ldp_msg_header *ldp_msg_header;
....
649.            msg_tptr=tptr+sizeof(struct ldp_msg_header);
```

Use of Sizeof On a Pointer Type\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=537
Status	New

	Source	Destination
File	PF_RING/print-ldp.c	PF_RING/print-ldp.c
Line	573	650
Object	ldp_msg_header	sizeof

Code Snippet

File Name PF_RING/print-ldp.c

Method ldp_pdu_print(netdissect_options *ndo,

```
....
573.      const struct ldp_msg_header *ldp_msg_header;
....
650.          msg_tlen=msg_len-(sizeof(struct ldp_msg_header)-4); /*
Type & Length fields not included */
```

Use of Sizeof On a Pointer Type\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=538
Status	New

	Source	Destination
File	PF_RING/print-ldp.c	PF_RING/print-ldp.c
Line	573	691
Object	ldp_msg_header	sizeof

Code Snippet

File Name PF_RING/print-ldp.c

Method ldp_pdu_print(netdissect_options *ndo,

```
....
573.      const struct ldp_msg_header *ldp_msg_header;
....
691.              print_unknown_data(ndo, tptr+sizeof(struct
ldp_msg_header), "\n\t  ",
```

Use of Sizeof On a Pointer Type\Path 24:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=539
Status	New

	Source	Destination
File	PF_RING/print-ldp.c	PF_RING/print-ldp.c
Line	250	277
Object	ldp_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-ldp.c
Method ldp_tlv_print(netdissect_options *ndo,

```
....
250.      const struct ldp_tlv_header *ldp_tlv_header;
....
277.      tptr+=sizeof(struct ldp_tlv_header);
```

Use of Sizeof On a Pointer Type\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=540
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	625
Object	lspping_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-lspping.c
Method lspping_print(netdissect_options *ndo,

```
....
494.      const struct lspping_tlv_header *lspping_tlv_header;
....
625.              tptr+=sizeof(struct lspping_tlv_header);
```

Use of Sizeof On a Pointer Type\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=541
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	609
Object	lspping_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-lspping.c
Method lspping_print(netdissect_options *ndo,

```
....
494.        const struct lspping_tlv_header *lspping_tlv_header;
....
609.            if (tlen < sizeof(struct lspping_tlv_header))
```

Use of Sizeof On a Pointer Type\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=542
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	626
Object	lspping_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-lspping.c
Method lspping_print(netdissect_options *ndo,

```
....
494.        const struct lspping_tlv_header *lspping_tlv_header;
....
626.                tlen-=sizeof(struct lspping_tlv_header);
```

Use of Sizeof On a Pointer Type\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=543
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	630

Object	lspping_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-lspping.c
Method lspping_print(netdissect_options *ndo,

```
....
494.        const struct lspping_tlv_header *lspping_tlv_header;
....
630.            tlv_tptr=tptr+sizeof(struct lspping_tlv_header);
```

Use of Sizeof On a Pointer Type\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=544
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	634
Object	lspping_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-lspping.c
Method lspping_print(netdissect_options *ndo,

```
....
494.        const struct lspping_tlv_header *lspping_tlv_header;
....
634.            if (tlen < lspping_tlv_len+sizeof(struct
lspping_tlv_header))
```

Use of Sizeof On a Pointer Type\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=545
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	644
Object	lspping_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-lspping.c

Method	lspping_print(netdissect_options *ndo,

```
....
494.      const struct lspping_tlv_header *lspping_tlv_header;
....
644.                  if (tlv_tlen < sizeof(struct lspping_tlv_header))
{
```

Use of Sizeof On a Pointer Type\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=546
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	654
Object	lspping_tlv_header	sizeof

Code Snippet	
File Name	PF_RING/print-lspping.c
Method	lspping_print(netdissect_options *ndo,

```
....
494.      const struct lspping_tlv_header *lspping_tlv_header;
....
654.                  subtlv_tptr=tlv_tptr+sizeof(struct
lspping_tlv_header);
```

Use of Sizeof On a Pointer Type\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=547
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	657
Object	lspping_tlv_header	sizeof

Code Snippet	
File Name	PF_RING/print-lspping.c
Method	lspping_print(netdissect_options *ndo,

```
....
494.      const struct lspping_tlv_header *lspping_tlv_header;
....
657.                     if (tlv_tlen < lspping_subtlv_len+sizeof(struct
lspping_tlv_header)) {
```

Use of Sizeof On a Pointer Type\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=548
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	863
Object	lspping_tlv_header	sizeof

Code Snippet

File Name	PF_RING/print-lspping.c
Method	lspping_print(netdissect_options *ndo,

```
....
494.      const struct lspping_tlv_header *lspping_tlv_header;
....
863.                        print_unknown_data(ndo, tlv_tptr+sizeof(struct
lspping_tlv_header),
```

Use of Sizeof On a Pointer Type\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=549
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	871
Object	lspping_tlv_header	sizeof

Code Snippet

File Name	PF_RING/print-lspping.c
Method	lspping_print(netdissect_options *ndo,

```
....
494.       const struct lspping_tlv_header *lspping_tlv_header;
....
871.                          if (tlv_tlen <
lspping_subtlv_len+sizeof(struct lspping_tlv_header)) {
```

Use of Sizeof On a Pointer Type\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=550
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	877
Object	lspping_tlv_header	sizeof

Code Snippet

File Name	PF_RING/print-lspping.c
Method	lspping_print(netdissect_options *ndo,

```
....
494.       const struct lspping_tlv_header *lspping_tlv_header;
....
877.                     tlv_tlen-=lspping_subtlv_len+sizeof(struct
lspping_tlv_header);
```

Use of Sizeof On a Pointer Type\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=551
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	1062
Object	lspping_tlv_header	sizeof

Code Snippet

File Name	PF_RING/print-lspping.c
Method	lspping_print(netdissect_options *ndo,

```
....
494.      const struct lspping_tlv_header *lspping_tlv_header;
....
1062.               print_unknown_data(ndo, tptr+sizeof(struct
lspping_tlv_header), "\n\t    ",
```

Use of Sizeof On a Pointer Type\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=552
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	1070
Object	lspping_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-lspping.c
Method lspping_print(netdissect_options *ndo,

```
....
494.      const struct lspping_tlv_header *lspping_tlv_header;
....
1070.               if (tlen < lspping_tlv_len+sizeof(struct
lspping_tlv_header))
```

Use of Sizeof On a Pointer Type\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=553
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	1074
Object	lspping_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-lspping.c
Method lspping_print(netdissect_options *ndo,

```
....
494.        const struct lspping_tlv_header *lspping_tlv_header;
....
1074.            tptr+=lspping_tlv_len+sizeof(struct lspping_tlv_header);
```

Use of Sizeof On a Pointer Type\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=554
Status	New

	Source	Destination
File	PF_RING/print-lspping.c	PF_RING/print-lspping.c
Line	494	1075
Object	lspping_tlv_header	sizeof

Code Snippet
File Name PF_RING/print-lspping.c
Method lspping_print(netdissect_options *ndo,

```
....
494.        const struct lspping_tlv_header *lspping_tlv_header;
....
1075.            tlen-=lspping_tlv_len+sizeof(struct lspping_tlv_header);
```

Use of Sizeof On a Pointer Type\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=555
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2710	2710
Object	sizeof	sizeof

Code Snippet
File Name PF_RING/optimize.c
Method convert_code_r(conv_state_t *conv_state, struct icode *ic, struct block *p)

```
....
2710.              offset = (struct slist **)calloc(slen, sizeof(struct
slist *));
```

Improper Resource Access Authorization

Query Path:
CPP\Cx\CPP Low Visibility\Improper Resource Access Authorization Version:1

Categories

FISMA 2014: Identification And Authentication
NIST SP 800-53: AC-3 Access Enforcement (P1)
OWASP Top 10 2017: A2-Broken Authentication

Description

Improper Resource Access Authorization\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=480
Status	New

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	496	496
Object	fgets	fgets

Code Snippet
File Name PF_RING/pfutils.c
Method int busid2node(int slot, int bus, int device, int function) {

```
....
496.       if (fgets(data, sizeof(data), fd) != NULL)
```

Improper Resource Access Authorization\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=481
Status	New

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	675	675
Object	fgetc	fgetc

Code Snippet
File Name PF_RING/pfutils.c
Method int read_packet_hex(u_char *buf, int buf_len) {

```
....
675.     while ((d = fgetc(stdin)) != EOF) {
```

Improper Resource Access Authorization\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=482
Status	New

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	496	496
Object	data	data

Code Snippet

File Name PF_RING/pfutils.c

Method int busid2node(int slot, int bus, int device, int function) {

```
....
496.      if (fgets(data, sizeof(data), fd) != NULL)
```

Improper Resource Access Authorization\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=483
Status	New

	Source	Destination
File	PF_RING/fttest.c	PF_RING/fttest.c
Line	300	300
Object	fprintf	fprintf

Code Snippet

File Name PF_RING/fttest.c

Method int main(int argc, char* argv[]) {

```
....
300.      fprintf(stderr, "pfring_ft_create_table error\n");
```

Improper Resource Access Authorization\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=484
Status	New

	Source	Destination
File	PF_RING/fttest.c	PF_RING/fttest.c
Line	138	138
Object	fprintf	fprintf

Code Snippet
File Name PF_RING/fttest.c
Method void sigproc(int sig) {

```
....
138.    fprintf(stderr, "Leaving...\n");
```

Improper Resource Access Authorization\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=485
Status	New

	Source	Destination
File	PF_RING/fttest.c	PF_RING/fttest.c
Line	221	221
Object	fprintf	fprintf

Code Snippet
File Name PF_RING/fttest.c
Method void packet_consumer() {

```
....
221.      fprintf(stderr, "Memory allocation failure\n");
```

Improper Resource Access Authorization\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=486
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2984	2984
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/optimize.c
Method	dot_dump_node(struct icode *ic, struct block *block, struct bpf_program *prog,

```
....
2984.       fprintf(out, "\tblock%u [shape=ellipse, id=\"block-%u\"
label=\"BLOCK%u\\n", block->id, block->id, block->id);
```

Improper Resource Access Authorization\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=487
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2986	2986
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/optimize.c
Method	dot_dump_node(struct icode *ic, struct block *block, struct bpf_program *prog,

```
....
2986.               fprintf(out, "\\n%s", bpf_image(prog->bf_insns + i,
i));
```

Improper Resource Access Authorization\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=488
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2988	2988
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/optimize.c
Method	dot_dump_node(struct icode *ic, struct block *block, struct bpf_program *prog,

```
....
2988.       fprintf(out, "\" tooltip=\"");
```

Improper Resource Access Authorization\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=489
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2991	2991
Object	fprintf	fprintf

Code Snippet
File Name PF_RING/optimize.c
Method dot_dump_node(struct icode *ic, struct block *block, struct bpf_program *prog,

```
....
2991.                   fprintf(out, "val[%d]=%d ", i, block->val[i]);
```

Improper Resource Access Authorization\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=490
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2992	2992
Object	fprintf	fprintf

Code Snippet
File Name PF_RING/optimize.c
Method dot_dump_node(struct icode *ic, struct block *block, struct bpf_program *prog,

```
....
2992.          fprintf(out, "val[A]=%d ", block->val[A_ATOM]);
```

Improper Resource Access Authorization\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=491
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2993	2993
Object	fprintf	fprintf

Code Snippet
File Name PF_RING/optimize.c
Method dot_dump_node(struct icode *ic, struct block *block, struct bpf_program *prog,

```
....
2993.        fprintf(out, "val[X]=%d", block->val[X_ATOM]);
```

Improper Resource Access Authorization\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=492
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2994	2994
Object	fprintf	fprintf

Code Snippet
File Name PF_RING/optimize.c
Method dot_dump_node(struct icode *ic, struct block *block, struct bpf_program *prog,

```
....
2994.        fprintf(out, "\"");
```

Improper Resource Access Authorization\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=493
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2996	2996
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/optimize.c
Method	dot_dump_node(struct icode *ic, struct block *block, struct bpf_program *prog,

```
....
2996.               fprintf(out, ", peripheries=2");
```

Improper Resource Access Authorization\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=494
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2997	2997
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/optimize.c
Method	dot_dump_node(struct icode *ic, struct block *block, struct bpf_program *prog,

```
....
2997.        fprintf(out, "];\n");
```

Improper Resource Access Authorization\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=495
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	3011	3011
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/optimize.c
Method	dot_dump_edge(struct icode *ic, struct block *block, FILE *out)

```
....
3011.               fprintf(out, "\t\"block%u\":se -> \"block%u\":n
[label=\"T\"]; \n",
```

Improper Resource Access Authorization\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=496
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	3013	3013
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/optimize.c
Method	dot_dump_edge(struct icode *ic, struct block *block, FILE *out)

```
....
3013.            fprintf(out, "\t\"block%u\":sw -> \"block%u\":n
[label=\"F\"]; \n",
```

Improper Resource Access Authorization\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=497
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	3050	3050
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/optimize.c
Method	dot_dump(struct icode *ic, char *errbuf)

```
....
3050.        fprintf(out, "digraph BPF {\n");
```

Improper Resource Access Authorization\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=498
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	3055	3055
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/optimize.c
Method	dot_dump(struct icode *ic, char *errbuf)

```
....
3055.        fprintf(out, "}\n");
```

Improper Resource Access Authorization\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=499
Status	New

	Source	Destination
File	PF_RING/parsenfsfh.c	PF_RING/parsenfsfh.c
Line	400	400
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/parsenfsfh.c
Method	Parse_fh(netdissect_options *ndo, const unsigned char *fh, u_int len,

```
....
400.                (void)fprintf(stderr, "%x.", GET_U_1(fhp + i));
```

Improper Resource Access Authorization\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=500
Status	New

	Source	Destination
File	PF_RING/parsenfsfh.c	PF_RING/parsenfsfh.c
Line	401	401
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/parsenfsfh.c
Method	Parse_fh(netdissect_options *ndo, const unsigned char *fh, u_int len,

```
....
401.            (void)fprintf(stderr, "\n");
```

Improper Resource Access Authorization\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=501
Status	New

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	356	356
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/pfutils.c
Method	int drop_privileges(const char *username) {

```
....
356.      fprintf(stderr, "privileges are not dropped as we're not
superuser\n");
```

Improper Resource Access Authorization\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=502
Status	New

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	369	369
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/pfutils.c
Method	int drop_privileges(const char *username) {

```
....
369.        fprintf(stderr, "unable to drop privileges [%s]\n",
strerror(errno));
```

Improper Resource Access Authorization\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=503
Status	New

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	372	372
Object	fprintf	fprintf

Code Snippet
File Name PF_RING/pfutils.c
Method int drop_privileges(const char *username) {

```
....
372.        fprintf(stderr, "user changed to %s\n", username);
```

Improper Resource Access Authorization\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=504
Status	New

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	375	375
Object	fprintf	fprintf

Code Snippet
File Name PF_RING/pfutils.c
Method int drop_privileges(const char *username) {

```
....
375.      fprintf(stderr, "unable to locate user %s\n", username);
```

Improper Resource Access Authorization\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=505
Status	New

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	393	393
Object	fprintf	fprintf

Code Snippet
File Name   PF_RING/pfutils.c
Method   void create_pid_file(char *pidFile) {

```
....
393.       fprintf(stderr, "unable to create pid file %s: %s\n", pidFile,
strerror(errno));
```

Improper Resource Access Authorization\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=506
Status	New

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	397	397
Object	fprintf	fprintf

Code Snippet
File Name   PF_RING/pfutils.c
Method   void create_pid_file(char *pidFile) {

```
....
397.     fprintf(fp, "%d\n", getpid());
```

Improper Resource Access Authorization\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=507
Status	New

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	534	534
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/pfutils.c
Method	int bindthread2core(pthread_t thread_id, int core_id) {

```
....
534.       fprintf(stderr, "Error while binding to core %u: errno=%i\n",
core_id, s);
```

Improper Resource Access Authorization\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=508
Status	New

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	618	618
Object	fprintf	fprintf

Code Snippet

File Name	PF_RING/pfutils.c
Method	void trace(int trace_level, char *file, int line, char * format, ...) {

```
....
618.     fprintf(out_file, "%s\n", out_buf);
```

Unchecked Array Index

Query Path:

CPP\Cx\CPP Low Visibility\Unchecked Array Index Version:1

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)

Description

Unchecked Array Index\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=556
Status	New

	Source	Destination
File	PF_RING/ice_idc.c	PF_RING/ice_idc.c
Line	211	211
Object	i	i

Code Snippet
File Name PF_RING/ice_idc.c
Method ice_alloc_rdma_qsets(struct iidc_core_dev_info *cdev_info,

```
....
211.                max_rdmaqs[i] = 0;
```

Unchecked Array Index\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=557
Status	New

	Source	Destination
File	PF_RING/ice_idc.c	PF_RING/ice_idc.c
Line	235	235
Object	tc	tc

Code Snippet
File Name PF_RING/ice_idc.c
Method ice_alloc_rdma_qsets(struct iidc_core_dev_info *cdev_info,

```
....
235.         vsi->qset_handle[qset->tc] = qset->qs_handle;
```

Unchecked Array Index\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=558
Status	New

	Source	Destination
File	PF_RING/ice_idc.c	PF_RING/ice_idc.c
Line	278	278
Object	tc	tc

Code Snippet
File Name PF_RING/ice_idc.c
Method ice_free_rdma_qsets(struct iidc_core_dev_info *cdev_info,

```
....
278.         vsi->qset_handle[qset->tc] = 0;
```

Unchecked Array Index\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=559
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	396	396
Object	level	level

Code Snippet
File Name PF_RING/optimize.c
Method find_levels_r(opt_state_t *opt_state, struct icode *ic, struct block *b)

```
....
396.        opt_state->levels[level] = b;
```

Unchecked Array Index\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=560
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	446	446
Object	dom	dom

Code Snippet
File Name PF_RING/optimize.c
Method find_dom(opt_state_t *opt_state, struct block *root)

```
....
446.                    SET_INSERT(b->dom, b->id);
```

Unchecked Array Index\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=561
Status	New

	Source	Destination

File	PF_RING/optimize.c	PF_RING/optimize.c
Line	458	458
Object	edom	edom

Code Snippet
File Name PF_RING/optimize.c
Method propedom(opt_state_t *opt_state, struct edge *ep)

```
....
458.         SET_INSERT(ep->edom, ep->id);
```

Unchecked Array Index\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=562
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	519	519
Object	closure	closure

Code Snippet
File Name PF_RING/optimize.c
Method find_closure(opt_state_t *opt_state, struct block *root)

```
....
519.                     SET_INSERT(b->closure, b->id);
```

Unchecked Array Index\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=563
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	762	762
Object	hash	hash

Code Snippet
File Name PF_RING/optimize.c
Method F(opt_state_t *opt_state, int code, bpf_u_int32 v0, bpf_u_int32 v1)

```
....
762.        opt_state->hashtbl[hash] = p;
```

Unchecked Array Index\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=564
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	2474	2474
Object	n	n

Code Snippet
File Name PF_RING/optimize.c
Method number_blks_r(opt_state_t *opt_state, struct icode *ic, struct block *p)

```
....
2474.        opt_state->blocks[n] = p;
```

Unchecked Array Index\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=565
Status	New

	Source	Destination
File	PF_RING/print-bootp.c	PF_RING/print-bootp.c
Line	1071	1071
Object	i	i

Code Snippet
File Name PF_RING/print-bootp.c
Method client_fqdn_flags(u_int flags)

```
....
1071.        buf[i] = '\0';
```

Unchecked Array Index\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=566
Status	New

	Source	Destination
File	PF_RING/print-rpki-rtr.c	PF_RING/print-rpki-rtr.c
Line	136	136
Object	idx	idx

Code Snippet
File Name PF_RING/print-rpki-rtr.c
Method indent_string (u_int indent)

```
....
136.        buf[idx] = '\0';
```

Unchecked Array Index\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=567
Status	New

	Source	Destination
File	PF_RING/print-rpki-rtr.c	PF_RING/print-rpki-rtr.c
Line	148	148
Object	idx	idx

Code Snippet
File Name PF_RING/print-rpki-rtr.c
Method indent_string (u_int indent)

```
....
148.        buf[idx] = '\n';
```

Unchecked Array Index\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=568
Status	New

	Source	Destination
File	PF_RING/print-rpki-rtr.c	PF_RING/print-rpki-rtr.c
Line	152	152

Object	idx	idx

Code Snippet
File Name PF_RING/print-rpki-rtr.c
Method indent_string (u_int indent)

```
....
152.        buf[idx] = '\t';
```

Unchecked Array Index\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=569
Status	New

	Source	Destination
File	PF_RING/print-rpki-rtr.c	PF_RING/print-rpki-rtr.c
Line	158	158
Object	idx	idx

Code Snippet
File Name PF_RING/print-rpki-rtr.c
Method indent_string (u_int indent)

```
....
158.        buf[idx] = ' ';
```

Unchecked Array Index\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=570
Status	New

	Source	Destination
File	PF_RING/print-rpki-rtr.c	PF_RING/print-rpki-rtr.c
Line	166	166
Object	idx	idx

Code Snippet
File Name PF_RING/print-rpki-rtr.c
Method indent_string (u_int indent)

```
....
166.        buf[idx] = '\0';
```

Unchecked Array Index\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=571
Status	New

	Source	Destination
File	PF_RING/print-rx.c	PF_RING/print-rx.c
Line	1090	1090
Object	i	i

Code Snippet
File Name PF_RING/print-rx.c
Method fs_reply_print(netdissect_options *ndo,

```
....
1090.                       a[i] = '\0';
```

Sizeof Pointer Argument

Query Path:
CPP\Cx\CPP Low Visibility\Sizeof Pointer Argument Version:0

Description

Sizeof Pointer Argument\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=316
Status	New

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3181	3182
Object	Pointer	sizeof

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_ipv6_addr(u8 *pkt, int offset, __be32 *addr)

```
....
3181.               memcpy(pkt + offset + idx * sizeof(*addr), &addr[idx],
3182.                       sizeof(*addr));
```

Sizeof Pointer Argument\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=317
Status	New

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3182	3182
Object	Pointer	sizeof

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_ipv6_addr(u8 *pkt, int offset, __be32 *addr)

```
....
3182.                           sizeof(*addr));
```

Sizeof Pointer Argument\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=318
Status	New

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3181	3182
Object	Pointer	sizeof

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_ipv6_addr(u8 *pkt, int offset, __be32 *addr)

```
....
3181.                   memcpy(pkt + offset + idx * sizeof(*addr), &addr[idx],
3182.                           sizeof(*addr));
```

Sizeof Pointer Argument\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=319
Status	New

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3182	3182
Object	Pointer	sizeof

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_ipv6_addr(u8 *pkt, int offset, __be32 *addr)

```
....
3182.                           sizeof(*addr));
```

Sizeof Pointer Argument\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=320
Status	New

	Source	Destination
File	PF_RING/util-print.c	PF_RING/util-print.c
Line	643	643
Object	bitmasks	sizeof

Code Snippet
File Name PF_RING/util-print.c
Method mask62plen(const u_char *mask)

```
....
643.                for (bits = 0; bits < (sizeof (bitmasks) / sizeof
(bitmasks[0])); bits++) {
```

Sizeof Pointer Argument\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=321
Status	New

	Source	Destination
File	PF_RING/util-print.c	PF_RING/util-print.c
Line	643	643
Object	bitmasks	sizeof

Code Snippet	
File Name	PF_RING/util-print.c
Method	mask62plen(const u_char *mask)

```
....
643.                for (bits = 0; bits < (sizeof (bitmasks) / sizeof
(bitmasks[0])); bits++) {
```

Sizeof Pointer Argument\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=322
Status	New

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3181	3181
Object	Pointer	sizeof

Code Snippet	
File Name	PF_RING/ice_fdir.c
Method	static void ice_pkt_insert_ipv6_addr(u8 *pkt, int offset, __be32 *addr)

```
....
3181.               memcpy(pkt + offset + idx * sizeof(*addr), &addr[idx],
```

Sizeof Pointer Argument\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=323
Status	New

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3182	3181
Object	Pointer	sizeof

Code Snippet	
File Name	PF_RING/ice_fdir.c
Method	static void ice_pkt_insert_ipv6_addr(u8 *pkt, int offset, __be32 *addr)

```
....
3182.                     sizeof(*addr));
....
3181.             memcpy(pkt + offset + idx * sizeof(*addr), &addr[idx],
```

Sizeof Pointer Argument\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=324
Status	New

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3181	3181
Object	Pointer	sizeof

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_ipv6_addr(u8 *pkt, int offset, __be32 *addr)

```
....
3181.             memcpy(pkt + offset + idx * sizeof(*addr), &addr[idx],
```

Sizeof Pointer Argument\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=325
Status	New

	Source	Destination
File	PF_RING/ice_fdir.c	PF_RING/ice_fdir.c
Line	3182	3181
Object	Pointer	sizeof

Code Snippet
File Name PF_RING/ice_fdir.c
Method static void ice_pkt_insert_ipv6_addr(u8 *pkt, int offset, __be32 *addr)

```
....
3182.                     sizeof(*addr));
....
3181.             memcpy(pkt + offset + idx * sizeof(*addr), &addr[idx],
```

Reliance on DNS Lookups in a Decision

Query Path:
CPP\Cx\CPP Low Visibility\Reliance on DNS Lookups in a Decision Version:0

Categories

FISMA 2014: Identification And Authentication
NIST SP 800-53: SC-23 Session Authenticity (P1)

Description

Reliance on DNS Lookups in a Decision\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=160
Status	New

The win32_gethostbyaddr method performs a reverse DNS lookup with gethostbyaddr, at line 149 of PF_RING/addrtoname.c. The application then makes a security decision, dotp, in PF_RING/addrtoname.c line 279, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	159	320
Object	gethostbyaddr	dotp

Code Snippet
File Name PF_RING/addrtoname.c
Method win32_gethostbyaddr(const char *addr, int len, int type)

```
....
159.            return gethostbyaddr(addr, len, type);
```

File Name PF_RING/addrtoname.c

Method ipaddr_string(netdissect_options *ndo, const u_char *ap)

```
....
320.                        if (dotp)
```

Reliance on DNS Lookups in a Decision\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=161
Status	New

The win32_gethostbyaddr method performs a reverse DNS lookup with gethostbyaddr, at line 149 of PF_RING/addrtoname.c. The application then makes a security decision, name, in PF_RING/addrtoname.c line 279, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	159	314
Object	gethostbyaddr	name

Code Snippet

File Name PF_RING/addrtoname.c

Method win32_gethostbyaddr(const char *addr, int len, int type)

```
....
159.                return gethostbyaddr(addr, len, type);
```

File Name PF_RING/addrtoname.c

Method ipaddr_string(netdissect_options *ndo, const u_char *ap)

```
....
314.                      if (p->name == NULL)
```

Reliance on DNS Lookups in a Decision\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=162
Status	New

The win32_gethostbyaddr method performs a reverse DNS lookup with gethostbyaddr, at line 149 of PF_RING/addrtoname.c. The application then makes a security decision, ==, in PF_RING/addrtoname.c line 279, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	159	314
Object	gethostbyaddr	==

Code Snippet

File Name PF_RING/addrtoname.c

Method win32_gethostbyaddr(const char *addr, int len, int type)

```
....
159.                return gethostbyaddr(addr, len, type);
```

File Name PF_RING/addrtoname.c

Method ipaddr_string(netdissect_options *ndo, const u_char *ap)

```
....
314.                    if (p->name == NULL)
```

Reliance on DNS Lookups in a Decision\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=163
Status	New

The win32_gethostbyaddr method performs a reverse DNS lookup with gethostbyaddr, at line 149 of PF_RING/addrtoname.c. The application then makes a security decision, hp, in PF_RING/addrtoname.c line 279, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	159	310
Object	gethostbyaddr	hp

Code Snippet

File Name PF_RING/addrtoname.c

Method win32_gethostbyaddr(const char *addr, int len, int type)

```
....
159.            return gethostbyaddr(addr, len, type);
```

File Name PF_RING/addrtoname.c

Method ipaddr_string(netdissect_options *ndo, const u_char *ap)

```
....
310.            if (hp) {
```

Reliance on DNS Lookups in a Decision\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=164
Status	New

The win32_gethostbyaddr method performs a reverse DNS lookup with gethostbyaddr, at line 149 of PF_RING/addrtoname.c. The application then makes a security decision, dotp, in PF_RING/addrtoname.c line 338, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c

Line	159	383
Object	gethostbyaddr	dotp

Code Snippet
File Name PF_RING/addrtoname.c
Method win32_gethostbyaddr(const char *addr, int len, int type)

```
....
159.            return gethostbyaddr(addr, len, type);
```

File Name PF_RING/addrtoname.c

Method ip6addr_string(netdissect_options *ndo, const u_char *ap)

```
....
383.                        if (dotp)
```

Reliance on DNS Lookups in a Decision\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=165
Status	New

The win32_gethostbyaddr method performs a reverse DNS lookup with gethostbyaddr, at line 149 of PF_RING/addrtoname.c. The application then makes a security decision, name, in PF_RING/addrtoname.c line 338, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	159	377
Object	gethostbyaddr	name

Code Snippet
File Name PF_RING/addrtoname.c
Method win32_gethostbyaddr(const char *addr, int len, int type)

```
....
159.            return gethostbyaddr(addr, len, type);
```

File Name PF_RING/addrtoname.c

Method ip6addr_string(netdissect_options *ndo, const u_char *ap)

```
....
377.                   if (p->name == NULL)
```

Reliance on DNS Lookups in a Decision\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=166
Status	New

The win32_gethostbyaddr method performs a reverse DNS lookup with gethostbyaddr, at line 149 of PF_RING/addrtoname.c. The application then makes a security decision, ==, in PF_RING/addrtoname.c line 338, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	159	377
Object	gethostbyaddr	==

Code Snippet
File Name PF_RING/addrtoname.c
Method win32_gethostbyaddr(const char *addr, int len, int type)

```
....
159.                return gethostbyaddr(addr, len, type);
```

File Name PF_RING/addrtoname.c
Method ip6addr_string(netdissect_options *ndo, const u_char *ap)

```
....
377.                       if (p->name == NULL)
```

Reliance on DNS Lookups in a Decision\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=167
Status	New

The win32_gethostbyaddr method performs a reverse DNS lookup with gethostbyaddr, at line 149 of PF_RING/addrtoname.c. The application then makes a security decision, hp, in PF_RING/addrtoname.c line 338, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	PF_RING/addrtoname.c	PF_RING/addrtoname.c
Line	159	373
Object	gethostbyaddr	hp

Code Snippet

File Name	PF_RING/addrtoname.c
Method	win32_gethostbyaddr(const char *addr, int len, int type)

```
....
159.              return gethostbyaddr(addr, len, type);
```

File Name	PF_RING/addrtoname.c
Method	ip6addr_string(netdissect_options *ndo, const u_char *ap)

```
....
373.              if (hp) {
```

TOCTOU

Query Path:

CPP\Cx\CPP Low Visibility\TOCTOU Version:1

Description

TOCTOU\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=513
Status	New

The create_pid_file method in PF_RING/pfutils.c file utilizes fopen that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	390	390
Object	fopen	fopen

Code Snippet

File Name	PF_RING/pfutils.c
Method	void create_pid_file(char *pidFile) {

```
....
390.     fp = fopen(pidFile, "w");
```

TOCTOU\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=514
Status	New

The busid2node method in PF_RING/pfutils.c file utilizes fopen that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	493	493
Object	fopen	fopen

Code Snippet
File Name PF_RING/pfutils.c
Method int busid2node(int slot, int bus, int device, int function) {

```
....
493.     if ((fd = fopen(path, "r")) != NULL) {
```

TOCTOU\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=515
Status	New

The i40e_client_subtask method in PF_RING/i40e_client.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	PF_RING/i40e_client.c	PF_RING/i40e_client.c
Line	447	447
Object	open	open

Code Snippet
File Name PF_RING/i40e_client.c
Method void i40e_client_subtask(struct i40e_pf *pf)

```
....
447.                    ret = client->ops->open(&cdev->lan_info,
client);
```

Incorrect Permission Assignment For Critical Resources

Query Path:
CPP\Cx\CPP Low Visibility\Incorrect Permission Assignment For Critical Resources Version:1

Categories

FISMA 2014: Access Control
NIST SP 800-53: AC-3 Access Enforcement (P1)
OWASP Top 10 2017: A2-Broken Authentication

Description

Incorrect Permission Assignment For Critical Resources\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=509
Status	New

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	390	390
Object	fp	fp

Code Snippet

File Name	PF_RING/pfutils.c
Method	void create_pid_file(char *pidFile) {

```
....
390.      fp = fopen(pidFile, "w");
```

Incorrect Permission Assignment For Critical Resources\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=510
Status	New

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	493	493
Object	fd	fd

Code Snippet

File Name	PF_RING/pfutils.c
Method	int busid2node(int slot, int bus, int device, int function) {

```
....
493.      if ((fd = fopen(path, "r")) != NULL) {
```

Exposure of System Data to Unauthorized Control Sphere

Query Path:
CPP\Cx\CPP Low Visibility\Exposure of System Data to Unauthorized Control Sphere Version:1

Categories

FISMA 2014: Configuration Management
NIST SP 800-53: AC-3 Access Enforcement (P1)

Description

Exposure of System Data to Unauthorized Control Sphere\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=511
Status	New

The system data read by drop_privileges in the file PF_RING/pfutils.c at line 352 is potentially exposed by drop_privileges found in PF_RING/pfutils.c at line 352.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	369	369
Object	errno	fprintf

Code Snippet
File Name PF_RING/pfutils.c
Method int drop_privileges(const char *username) {

```
....
369.          fprintf(stderr, "unable to drop privileges [%s]\n", 
strerror(errno));
```

Exposure of System Data to Unauthorized Control Sphere\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=512
Status	New

The system data read by create_pid_file in the file PF_RING/pfutils.c at line 385 is potentially exposed by create_pid_file found in PF_RING/pfutils.c at line 385.

	Source	Destination
File	PF_RING/pfutils.c	PF_RING/pfutils.c
Line	393	393
Object	errno	fprintf

Code Snippet
File Name PF_RING/pfutils.c
Method void create_pid_file(char *pidFile) {

```
....
393.        fprintf(stderr, "unable to create pid file %s: %s\n", pidFile, 
strerror(errno));
```

Inconsistent Implementations

Query Path:
CPP\Cx\CPP Low Visibility\Inconsistent Implementations Version:0

Description

Inconsistent Implementations\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=3
Status	New

	Source	Destination
File	PF_RING/fttest.c	PF_RING/fttest.c
Line	266	266
Object	getopt	getopt

Code Snippet

File Name	PF_RING/fttest.c
Method	int main(int argc, char* argv[]) {

```
....
266.     while ((c = getopt(argc,argv,"g:hqvS:7")) != '?') {
```

Arithmenic Operation On Boolean

Query Path:
CPP\Cx\CPP Low Visibility\Arithmenic Operation On Boolean Version:1

Categories

FISMA 2014: Audit And Accountability
NIST SP 800-53: SC-5 Denial of Service Protection (P1)

Description

Arithmenic Operation On Boolean\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1020018&projectid=20014&pathid=315
Status	New

	Source	Destination
File	PF_RING/optimize.c	PF_RING/optimize.c
Line	391	391
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name	PF_RING/optimize.c
Method	find_levels_r(opt_state_t *opt_state, struct icode *ic, struct block *b)

```
....
391.                level = MAX(JT(b)->level, JF(b)->level) + 1;
```

Buffer Overflow LongString

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

CPP

Overflowing Buffers

```
const int BUFFER_SIZE = 10;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
{
    strcpy(buffer, inputString);
}
```

Checked Buffers

```
const int BUFFER_SIZE = 10;
const int MAX_INPUT_SIZE = 256;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
{
    if (strnlen(inputString, MAX_INPUT_SIZE) < sizeof(buffer))
    {
        strncpy(buffer, inputString, sizeof(buffer));
    }
}
```

Divide By Zero

Risk

What might happen

When a program divides a number by zero, an exception will be raised. If this exception is not handled by the application, unexpected results may occur, including crashing the application. This can be considered a DoS (Denial of Service) attack, if an external user has control of the value of the denominator or can cause this error to occur.

Cause

How does it happen

The program receives an unexpected value, and uses it for division without filtering, validation, or verifying that the value is not zero. The application does not explicitly handle this error or prevent division by zero from occuring.

General Recommendations

How to avoid it

- Before dividing by an unknown value, validate the number and explicitly ensure it does not evaluate to zero.
- Validate all untrusted input from all sources, in particular verifying that it is not zero before dividing with it.
- Verify output of methods, calculations, dictionary lookups, and so on, and ensure it is not zero before dividing with the result.
- Ensure divide-by-zero errors are caught and handled appropriately.

Source Code Examples

Java

Divide by Zero

```
public float getAverage(HttpServletRequest req) {
      int total = Integer.parseInt(req.getParameter("total"));
      int count = Integer.parseInt(req.getParameter("count"));

      return total / count;
}
```

Checked Division

```
public float getAverage(HttpServletRequest req) {
      int total = Integer.parseInt(req.getParameter("total"));
      int count = Integer.parseInt(req.getParameter("count"));
      if (count > 0)
             return total / count;
      else
             return 0;
}
```

Buffer Overflow boundcpy WrongSizeParam

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

Wrong Size t Allocation

Risk

What might happen

Incorrect allocation of memory may result in unexpected behavior by either overwriting sections of memory with unexpected values. Under certain conditions where both an incorrect allocation of memory and the values being written can be controlled by an attacker, such an issue may result in execution of malicious code.

Cause

How does it happen

Some memory allocation functions require a size value to be provided as a parameter. The allocated size should be derived from the provided value, by providing the length value of the intended source, multiplied by the size of that length. Failure to perform the correct arithmetic to obtain the exact size of the value will likely result in the source overflowing its destination.

General Recommendations

How to avoid it

- Always perform the correct arithmetic to determine size.
- Specifically for memory allocation, calculate the allocation size from the allocation source:
    - Derive the size value from the length of intended source to determine the amount of units to be processed.
    - Always programmatically consider the size of the each unit and their conversion to memory units - for example, by using sizeof() on the unit's type.
    - Memory allocation should be a multiplication of the amount of units being written, times the size of each unit.

Source Code Examples

Char Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
- Prefer promoting the target variable to a large enough data type.
- If downcasting is necessary, always check that values are valid and in range of the target type, before casting

Source Code Examples

CPP

Unsafe Downsize Casting

```
int unsafe_addition(short op1, int op2) {

    // op2 gets forced from int into a short
    short total = op1 + op2;

    return total;
}
```

Safer Use of Proper Data Types

```
int safe_addition(short op1, int op2) {

    // total variable is of type int, the largest type that is needed
    int total = 0;

    // check if total will overflow available integer size
    if (INT_MAX - abs(op2) > op1)
    {
        total = op1 + op2;
    }
    else
    {
        // instead of overflow, saturate (but this is not always a good thing)
        total = INT_MAX
    }

    return total;
}
```

Integer Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
- Prefer promoting the target variable to a large enough data type.
- If downcasting is necessary, always check that values are valid and in range of the target type, before casting

Source Code Examples

Dangerous Functions

Risk

What might happen

Use of dangerous functions may expose varying risks associated with each particular function, with potential impact of improper usage of these functions varying significantly. The presence of such functions indicates a flaw in code maintenance policies and adherence to secure coding practices, in a way that has allowed introducing known dangerous code into the application.

Cause

How does it happen

A dangerous function has been identified within the code. Functions are often deemed dangerous to use for numerous reasons, as there are different sets of vulnerabilities associated with usage of such functions. For example, some string copy and concatenation functions are vulnerable to Buffer Overflow, Memory Disclosure, Denial of Service and more. Use of these functions is not recommended.

General Recommendations

How to avoid it

- Deploy a secure and recommended alternative to any functions that were identified as dangerous.
    - If no secure alternative is found, conduct further researching and testing to identify whether current usage successfully sanitizes and verifies values, and thus successfully avoids the use-cases for whom the function is indeed dangerous
- Conduct a periodical review of methods that are in use, to ensure that all external libraries and built-in functions are up-to-date and whose use has not been excluded from best secure coding practices.

Source Code Examples

CPP

Buffer Overflow in gets()

```
int main()

{

        char buf[10];

        printf("Please enter your name: ");
        gets(buf); // veryveryverylongname
        if (buf == ACCEPTED_NAME)
        {
                // Do something
        }
        return 0;
}
```

Safe reading from user

```c
int main()
{
      char buf[10];

      printf("Please enter your name: ");
      fgets(buf, sizeof(buf), stdin); //setting the amount of bytes to read
      if (buf == ACCEPTED_NAME)
      {
             //Do something
      }
      return 0;
}
```

Unsafe function for string copy

```c
int main(int argc, char* argv[])
{
      char buf[10];
      strcpy(buf, argv[1]); // overflow occurs when len(argv[1]) > 10 bytes

      return 0;
}
```

Safe string copy

```c
int main(int argc, char* argv[])
{
      char buf[10];
      strncpy(buf, argv[1], sizeof(buf));
      buf[9]= '\0'; //strncpy doesn't NULL terminates

      return 0;
}
```

Unsafe format string

```c
int main(int argc, char* argv[])
{
      printf(argv[1]); // If argv[1] contains a format token, such as %s,%x or %d, will cause
an access violation
      return 0;
}
```

Safe format string

```
int main(int argc, char* argv[])
{
      printf("%s", argv[1]); // Second parameter is not a formattable string

      return 0;
}
```

Double Free

Weakness ID: 415 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The product calls free() twice on the same memory address, potentially leading to modification of unexpected memory locations.

Extended Description

When a program calls free() twice with the same argument, the program's memory management data structures become corrupted. This corruption can cause the program to crash or, in some circumstances, cause two later calls to malloc() to return the same pointer. If malloc() returns the same value twice and the program later gives the attacker control over the data that is written into this doubly-allocated memory, the program becomes vulnerable to a buffer overflow attack.

Alternate Terms

Double-free

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Access Control	Doubly freeing memory may result in a write-what-where condition, allowing an attacker to execute arbitrary code.

Likelihood of Exploit

Low to Medium

Demonstrative Examples

Example 1

The following code shows a simple example of a double free vulnerability.

(Bad Code)

Example Language: ***C***

```
char* ptr = (char*)malloc (SIZE);
...
if (abrt) {
free(ptr);
}
...
free(ptr);
```

Double free vulnerabilities have two common (and sometimes overlapping) causes:

- Error conditions and other exceptional circumstances
- Confusion over which part of the program is responsible for freeing the memory

Although some double free vulnerabilities are not much more complicated than the previous example, most are spread out across hundreds of lines of code or even different files. Programmers seem particularly susceptible to freeing global variables more than once.

Example 2

While contrived, this code should be exploitable on Linux distributions which do not ship with heap-chunk check summing turned on.

(Bad Code)

Example Language: **C**

```
#include <stdio.h>
#include <unistd.h>
#define BUFSIZE1 512
#define BUFSIZE2 ((BUFSIZE1/2) - 8)

int main(int argc, char **argv) {
char *buf1R1;
char *buf2R1;
char *buf1R2;
buf1R1 = (char *) malloc(BUFSIZE2);
buf2R1 = (char *) malloc(BUFSIZE2);
free(buf1R1);
free(buf2R1);
buf1R2 = (char *) malloc(BUFSIZE1);
strncpy(buf1R2, argv[1], BUFSIZE1-1);
free(buf2R1);
free(buf1R2);
}
```

Observed Examples

Reference	Description
CVE-2004-0642	Double free resultant from certain error conditions.
CVE-2004-0772	Double free resultant from certain error conditions.
CVE-2005-1689	Double free resultant from certain error conditions.
CVE-2003-0545	Double free from invalid ASN.1 encoding.
CVE-2003-1048	Double free from malformed GIF.
CVE-2005-0891	Double free from malformed GIF.
CVE-2002-0059	Double free from malformed compressed data.

Potential Mitigations

Phase: Architecture and Design

Choose a language that provides automatic memory management.

Phase: Implementation

Ensure that each allocation is freed only once. After freeing a chunk, set the pointer to NULL to ensure the pointer cannot be freed again. In complicated error conditions, be sure that clean-up routines respect the state of allocation properly. If the language is object oriented, ensure that object destructors delete each chunk of memory only once.

Phase: Implementation

Use a static analysis tool to find double free instances.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Weakness Base	666	Operation on Resource in Wrong Phase of	**Research Concepts (primary)1000**

			Lifetime	
ChildOf	Weakness Class	675	Duplicate Operations on Resource	Research Concepts1000
ChildOf	Category	742	CERT C Secure Coding Section 08 - Memory Management (MEM)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
PeerOf	Weakness Base	123	Write-what-where Condition	Research Concepts1000
PeerOf	Weakness Base	416	Use After Free	Development Concepts699 Research Concepts1000
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**
PeerOf	Weakness Base	364	Signal Handler Race Condition	Research Concepts1000

Relationship Notes

This is usually resultant from another weakness, such as an unhandled error or race condition between threads. It could also be primary to weaknesses such as buffer overflows.

Affected Resources

- Memory

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			DFREE - Double-Free Vulnerability
7 Pernicious Kingdoms			Double Free
CLASP			Doubly freeing memory
CERT C Secure Coding	MEM00-C		Allocate and free memory in the same module, at the same level of abstraction
CERT C Secure Coding	MEM01-C		Store a new value in pointers immediately after free()
CERT C Secure Coding	MEM31-C		Free dynamically allocated memory exactly once

White Box Definitions

A weakness where code path has:

1. start statement that relinquishes a dynamically allocated memory resource
2. end statement that relinquishes the dynamically allocated memory resource

Maintenance Notes

It could be argued that Double Free would be most appropriately located as a child of "Use after Free", but "Use" and "Release" are considered to be distinct operations within vulnerability theory, therefore this is more accurately "Release of a Resource after Expiration or Release", which doesn't exist yet.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	PLOVER		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Potential Mitigations, Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Description, Maintenance Notes, Relationships, Other Notes, Relationship Notes, Taxonomy Mappings		
2008-11-24	CWE Content Team	MITRE	Internal

	updated Relationships, Taxonomy Mappings		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Other Notes		

BACK TO TOP

Failure to Release Memory Before Removing Last Reference ('Memory Leak')

Weakness ID: 401 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The software does not sufficiently track and release allocated memory after it has been used, which slowly consumes remaining memory.

Extended Description

This is often triggered by improper handling of malformed data or unexpectedly interrupted sessions.

Terminology Notes

"memory leak" has sometimes been used to describe other kinds of issues, e.g. for information leaks in which the contents of memory are inadvertently leaked (CVE-2003-0400 is one such example of this terminology conflict).

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Modes of Introduction

Memory leaks have two common and sometimes overlapping causes:

- Error conditions and other exceptional circumstances
- Confusion over which part of the program is responsible for freeing the memory

Common Consequences

Scope	Effect
Availability	Most memory leaks result in general software reliability problems, but if an attacker can intentionally trigger a memory leak, the attacker might be able to launch a denial of service attack (by crashing or hanging the program) or take advantage of other unexpected program behavior resulting from a low memory condition.

Likelihood of Exploit

Medium

Demonstrative Examples

Example 1

The following C function leaks a block of allocated memory if the call to read() fails to return the expected number of bytes:

(Bad Code)

Example Language: **C**

```
char* getBlock(int fd) {
char* buf = (char*) malloc(BLOCK_SIZE);
if (!buf) {
return NULL;
}
if (read(fd, buf, BLOCK_SIZE) != BLOCK_SIZE) {

return NULL;
}
return buf;
}
```

Example 2

Here the problem is that every time a connection is made, more memory is allocated. So if one just opened up more and more connections, eventually the machine would run out of memory.

(Bad Code)

Example Language: **C**

```
bar connection(){
foo = malloc(1024);
return foo;
}
endConnection(bar foo) {

free(foo);
}
int main() {

while(1) //thread 1
//On a connection
foo=connection(); //thread 2
//When the connection ends
endConnection(foo)
}
```

Observed Examples

Reference	Description
CVE-2005-3119	Memory leak because function does not free() an element of a data structure.
CVE-2004-0427	Memory leak when counter variable is not decremented.
CVE-2002-0574	Memory leak when counter variable is not decremented.
CVE-2005-3181	Kernel uses wrong function to release a data structure, preventing data from being properly tracked by other code.
CVE-2004-0222	Memory leak via unknown manipulations as part of protocol test suite.
CVE-2001-0136	Memory leak via a series of the same command.

Potential Mitigations

Pre-design: Use a language or compiler that performs automatic bounds checking.

Phase: Architecture and Design

Use an abstraction library to abstract away risky APIs. Not a complete solution.

Pre-design through Build: The Boehm-Demers-Weiser Garbage Collector or valgrind can be used to detect leaks in code. This is not a complete solution as it is not 100% effective.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	730	OWASP Top Ten 2004 Category A9 - Denial of Service	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Weakness Base	772	Missing Release of Resource after Effective	**Research Concepts (primary)1000**

			Lifetime	
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**
CanFollow	Weakness Class	390	Detection of Error Condition Without Action	Research Concepts1000

Relationship Notes

This is often a resultant weakness due to improper handling of malformed data or early termination of sessions.

Affected Resources

- Memory

Functional Areas

- Memory management

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			Memory leak
7 Pernicious Kingdoms			Memory Leak
CLASP			Failure to deallocate data
OWASP Top Ten 2004	A9	CWE More Specific	Denial of Service

White Box Definitions

A weakness where the code path has:

1. start statement that allocates dynamically allocated memory resource

2. end statement that loses identity of the dynamically allocated memory resource creating situation where dynamically allocated memory resource is never relinquished

Where "loses" is defined through the following scenarios:

1. identity of the dynamic allocated memory resource never obtained

2. the statement assigns another value to the data element that stored the identity of the dynamically allocated memory resource and there are no aliases of that data element

3. identity of the dynamic allocated memory resource obtained but never passed on to function for memory resource release

4. the data element that stored the identity of the dynamically allocated resource has reached the end of its scope at the statement and there are no aliases of that data element

References

J. Whittaker and H. Thompson. "How to Break Software Security". Addison Wesley. 2003.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	PLOVER		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, References, Relationship Notes, Taxonomy Mappings, Terminology Notes		
2008-10-14	CWE Content Team	MITRE	Internal
	updated Description		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Other Notes		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Name		
2009-07-17	KDM Analytics		External
	Improved the White Box Definition		

2009-07-27	CWE Content Team	MITRE	Internal
	updated White Box Definitions		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Modes of Introduction, Other Notes		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Memory Leak
2009-05-27	Failure to Release Memory Before Removing Last Reference (aka 'Memory Leak')

BACK TO TOP

Use of Uninitialized Pointer

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

Use of Zero Initialized Pointer

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

CPP

Explicit NULL Dereference

```
char * input = NULL;
printf("%s", input);
```

Implicit NULL Dereference

```
char * input;
printf("%s", input);
```

Java

Explicit Null Dereference

```
Object o = null;
out.println(o.getClass());
```

Wrong Memory Allocation

Risk

What might happen

Incorrect allocation of memory may result in unexpected behavior by either overwriting sections of memory with unexpected values. Under certain conditions where both an incorrect allocation of memory and the values being written can be controlled by an attacker, such an issue may result in execution of malicious code.

Cause

How does it happen

Some memory allocation functions require a size value to be provided as a parameter. The allocated size should be derived from the provided value, by providing the length value of the intended source, multiplied by the size of that length. Failure to perform the correct arithmetic to obtain the exact size of the value will likely result in the source overflowing its destination.

General Recommendations

How to avoid it

- Always perform the correct arithmetic to determine size.
- Specifically for memory allocation, calculate the allocation size from the allocation source:
    - Derive the size value from the length of intended source to determine the amount of units to be processed.
    - Always programmatically consider the size of the each unit and their conversion to memory units - for example, by using sizeof() on the unit's type.
    - Memory allocation should be a multiplication of the amount of units being written, times the size of each unit.

Source Code Examples

CPP

Allocating and Assigning Memory without Sizeof Arithmetic

```
int *ptr;
ptr = (int*)malloc(5);
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1;
}
```

Allocating and Assigning Memory with Sizeof Arithmetic

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1;
}
```

Incorrect Arithmetic of Multi-Byte String Allocation

```
wchar_t * dest;
dest = (wchar_t *)malloc(wcslen(source) + 1); // Would not crash for a short "source"
wcscpy((wchar_t *)dest, source);
wprintf(L"Dest: %s\r\n", dest);
```

Correct Arithmetic of Multi-Byte String Allocation

```
wchar_t * dest;
dest = (wchar_t *)malloc((wcslen(source) + 1) * sizeof(wchar_t));
wcscpy((wchar_t *)dest, source);
wprintf(L"Dest: %s\r\n", dest);
```

Inadequate Encryption Strength

Risk

What might happen

Using weak or outdated cryptography does not provide sufficient protection for sensitive data. An attacker that gains access to the encrypted data would likely be able to break the encryption, using either cryptanalysis or brute force attacks. Thus, the attacker would be able to steal user passwords and other personal data. This could lead to user impersonation or identity theft.

Cause

How does it happen

The application uses a weak algorithm, that is considered obselete since it is relatively easy to break. These obselete algorithms are vulnerable to several different kinds of attacks, including brute force.

General Recommendations

How to avoid it

Generic Guidance:

- Always use strong, modern algorithms for encryption, hashing, and so on.
- Do not use weak, outdated, or obsolete algorithms.
- Ensure you select the correct cryptographic mechanism according to the specific requirements.
- Passwords should be protected with a dedicated password protection scheme, such as bcrypt, scrypt, PBKDF2, or Argon2.

Specific Recommendations:

- Do not use SHA-1, MD5, or any other weak hash algorithm to protect passwords or personal data. Instead, use a stronger hash such as SHA-256 when a secure hash is required.
- Do not use DES, Triple-DES, RC2, or any other weak encryption algorithm to protect passwords or personal data. Instead, use a stronger encryption algorithm such as AES to protect personal data.
- Do not use weak encryption modes such as ECB, or rely on insecure defaults. Explicitly specify a stronger encryption mode, such as GCM.
- For symmetric encryption, use a key length of at least 256 bits.

Source Code Examples

Java

Weakly Hashed PII

```
string protectSSN(HttpServletRequest req) {
    string socialSecurityNum = req.getParameter("SocialSecurityNo");

    return DigestUtils.md5Hex(socialSecurityNum);
}
```

Stronger Hash for PII

```
string protectSSN(HttpServletRequest req) {
     string socialSecurityNum = req.getParameter("SocialSecurityNo");

     return DigestUtils.sha256Hex(socialSecurityNum);
}
```

Use of Function with Inconsistent Implementations

Weakness ID: 474 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The code uses a function that has inconsistent implementations across operating systems and versions, which might cause security-relevant portability problems.

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C: *(Often)*

PHP: *(Often)*

All

Potential Mitigations

Do not accept inconsistent behavior from the API specifications when the deviant behavior increase the risk level.

Other Notes

The behavior of functions in this category varies by operating system, and at times, even by operating system version. Implementation differences can include:

- Slight differences in the way parameters are interpreted leading to inconsistent results.

- Some implementations of the function carry significant security risks.

- The function might not be defined on all platforms.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Development Concepts (primary)699** **Seven Pernicious Kingdoms (primary)700** **Research Concepts (primary)1000**
ParentOf	Weakness Variant	589	Call to Non-ubiquitous API	**Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Inconsistent Implementations

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	7 Pernicious Kingdoms		Externally Mined

Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Potential Mitigations, Time of Introduction		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Relationships, Other Notes, Taxonomy Mappings		

Previous Entry Names	
Change Date	**Previous Entry Name**
2008-04-11	Inconsistent Implementations

BACK TO TOP

Unchecked Return Value

Risk

What might happen

A program that does not check function return values could cause the application to enter an undefined state. This could lead to unexpected behavior and unintended consequences, including inconsistent data, system crashes or other error-based exploits.

Cause

How does it happen

The application calls a system function, but does not receive or check the result of this funciton. These functions often return error codes in the result, or share other status codes with it's caller. The application simply ignores this result value, losing this vital information.

General Recommendations

How to avoid it

- Always check the result of any called function that returns a value, and verify the result is an expected value.

- Ensure the calling function responds to all possible return values.

- Expect runtime errors and handle them gracefully. Explicitly define a mechanism for handling unexpected errors.

Source Code Examples

CPP

Unchecked Memory Allocation

```
buff = (char*) malloc(size);
strncpy(buff, source, size);
```

Safer Memory Allocation

```
buff = (char*) malloc(size+1);
if (buff==NULL) exit(1);

strncpy(buff, source, size);
buff[size] = '\0';
```

Reliance on DNS Lookups in a Decision

Risk

What might happen

Relying on reverse DNS records, without verifying domain ownership via cryptographic certificates or protocols, is not a sufficient authentication mechanism. Basing any security decisions on the registered hostname could allow an external attacker to control the application flow. The attacker could possibly perform restricted operations, bypass access controls, and even spoof the user's identity, inject a bogus hostname into the security log, and possibly other logic attacks.

Cause

How does it happen

The application performs a reverse DNS resolution, based on the remote IP address, and performs a security check based on the returned hostname. However, it is relatively easy to spoof DNS names, or cause them to be misreported, depending on the context of the specific environment. If the remote server is controlled by the attacker, it can be configured to report a bogus hostname. Additionally, the attacker could also spoof the hostname if she controls the associated DNS server, or by attacking the legitimate DNS server, or by poisoning the server's DNS cache, or by modifying unprotected DNS traffic to the server. Regardless of the vector, a remote attacker can alter the detected network address, faking the authentication details.

General Recommendations

How to avoid it

- Do not rely on DNS records, network addresses, or system hostnames as a form of authentication, or any other security-related decision.
- Do not perform reverse DNS resolution over an unprotected protocol without record validation.
- Implement a proper authentication mechanism, such as passwords, cryptographic certificates, or public key digital signatures.
- Consider using proposed protocol extensions to cryptographically protect DNS, e.g. DNSSEC (though note the limited support and other drawbacks).

Source Code Examples

Java

Using Reverse DNS as Authentication

```
private boolean isInternalEmployee(ServletRequest req) {
      boolean isCompany = false;

      String ip = req.getRemoteAddr();
      InetAddress address = InetAddress.getByName(ip);

      if (address.getHostName().endsWith(COMPANYNAME)) {
             isCompany = true;
     }
      return isCompany;
}
```

Verify Authenticated User's Identity

```
private boolean isInternalEmployee(ServletRequest req) {
      boolean isCompany = false;

      Principal user = req.getUserPrincipal();
      if (user != null) {
      if (user.getName().startsWith(COMPANYDOMAIN + "\\")) {
             isCompany = true;
         }
     }
      return isCompany;
}
```

NULL Pointer Dereference

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

Indicator of Poor Code Quality

Weakness ID: 398 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The code has features that do not directly introduce a weakness or vulnerability, but indicate that the product has not been carefully developed or maintained.

Extended Description

Programs are more likely to be secure when good development practices are followed. If a program is complex, difficult to maintain, not portable, or shows evidence of neglect, then there is a higher likelihood that weaknesses are buried in the code.

Time of Introduction

- Architecture and Design
- Implementation

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	18	Source Code	**Development Concepts (primary)699**
ChildOf	Weakness Class	710	Coding Standards Violation	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	107	Struts: Unused Validation Form	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	110	Struts: Validator Without Form Field	**Research Concepts (primary)1000**
ParentOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ParentOf	Weakness Base	401	Failure to Release Memory Before Removing Last Reference ('Memory Leak')	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	404	Improper Resource Shutdown or Release	Development Concepts699 **Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Variant	415	Double Free	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	416	Use After Free	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Variant	457	Use of Uninitialized Variable	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	474	Use of Function with Inconsistent Implementations	**Development Concepts (primary)699** **Seven Pernicious Kingdoms (primary)700** **Research Concepts (primary)1000**
ParentOf	Weakness Base	475	Undefined Behavior for Input to API	**Development Concepts (primary)699** **Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	476	NULL Pointer	**Development**

			Dereference	**Concepts (primary)699** **Seven Pernicious Kingdoms (primary)700** **Research Concepts (primary)1000**
ParentOf	Weakness Base	477	Use of Obsolete Functions	**Development Concepts (primary)699** **Seven Pernicious Kingdoms (primary)700** **Research Concepts (primary)1000**
ParentOf	Weakness Variant	478	Missing Default Case in Switch Statement	**Development Concepts (primary)699**
ParentOf	Weakness Variant	479	Unsafe Function Call from a Signal Handler	**Development Concepts (primary)699**
ParentOf	Weakness Variant	483	Incorrect Block Delimitation	**Development Concepts (primary)699**
ParentOf	Weakness Base	484	Omitted Break Statement in Switch	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Variant	546	Suspicious Comment	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ParentOf	Weakness Variant	547	Use of Hard-coded, Security-relevant Constants	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ParentOf	Weakness Variant	561	Dead Code	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ParentOf	Weakness Base	562	Return of Stack Variable Address	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Variant	563	Unused Variable	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ParentOf	Category	569	Expression Issues	**Development Concepts (primary)699**
ParentOf	Weakness Variant	585	Empty Synchronized Block	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ParentOf	Weakness Variant	586	Explicit Call to Finalize()	**Development Concepts (primary)699**
ParentOf	Weakness Variant	617	Reachable Assertion	**Development Concepts (primary)699**
ParentOf	Weakness Base	676	Use of Potentially Dangerous Function	**Development Concepts (primary)699** **Research Concepts (primary)1000**
MemberOf	View	700	Seven Pernicious Kingdoms	**Seven Pernicious Kingdoms (primary)700**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name

7 Pernicious Kingdoms			Code Quality

Content History

Submissions

Submission Date	Submitter	Organization	Source
	7 Pernicious Kingdoms		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci updated Time of Introduction	Cigital	External
2008-09-08	CWE Content Team updated Description, Relationships, Taxonomy Mappings	MITRE	Internal
2009-10-29	CWE Content Team updated Relationships	MITRE	Internal

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Code Quality

BACK TO TOP

Use of sizeof() on a Pointer Type

Weakness ID: 467 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Integrity	This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(Bad Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(Good Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(Bad Code)

```

/* Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */

char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```

```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strncmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strncmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In AuthenticateUser(), because sizeof() is applied to a parameter with an array type, the sizeof() call might return 4 on many modern architectures. As a result, the strncmp() call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "sizeof(*pointer)" instead of "sizeof(pointer)", unless you intend to run sizeof() on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of sizeof() on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of sizeof(pointer) indicates a bug.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	465	Pointer Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	682	Incorrect Calculation	**Research Concepts (primary)1000**
ChildOf	Category	737	CERT C Secure Coding Section 03 - Expressions (EXP)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
CanPrecede	Weakness Base	131	Incorrect Calculation of Buffer Size	Research Concepts1000

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Use of sizeof() on a pointer type
CERT C Secure Coding	ARR01-C		Do not apply the sizeof operator to a pointer when taking the size of an array
CERT C Secure Coding	EXP01-C		Do not take the size of a pointer to determine the size of the pointed-to type

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator
2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type". <https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

BACK TO TOP

Improper Access Control (Authorization)

Weakness ID: 285 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The software does not perform or incorrectly performs access control checks across all potential execution paths.

Extended Description

When access control checks are not applied consistently - or not at all - users are able to access data or perform actions that they should not be allowed to perform. This can lead to a wide range of problems, including information leaks, denial of service, and arbitrary code execution.

Alternate Terms

AuthZ:	"AuthZ" is typically used as an abbreviation of "authorization" within the web application security community. It is also distinct from "AuthC," which is an abbreviation of "authentication." The use of "Auth" as an abbreviation is discouraged, since it could be used for either authentication or authorization.

Time of Introduction

- Architecture and Design
- Implementation
- Operation

Applicable Platforms

Languages

Language-independent

Technology Classes

Web-Server: *(Often)*

Database-Server: *(Often)*

Modes of Introduction

A developer may introduce authorization weaknesses because of a lack of understanding about the underlying technologies. For example, a developer may assume that attackers cannot modify certain inputs such as headers or cookies.

Authorization weaknesses may arise when a single-user application is ported to a multi-user environment.

Common Consequences

Scope	Effect
Confidentiality	An attacker could read sensitive data, either by reading the data directly from a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to read the data.
Integrity	An attacker could modify sensitive data, either by writing the data directly to a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to write the data.
Integrity	An attacker could gain privileges by modifying or reading critical data directly, or by accessing insufficiently-protected, privileged functionality.

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

Automated static analysis is useful for detecting commonly-used idioms for authorization. A tool may be able to analyze related configuration files, such as .htaccess in Apache web servers, or detect the usage of commonly-used authorization libraries.

Generally, automated static analysis tools have difficulty detecting custom authorization schemes. In addition, the software's design may include some functionality that is accessible to any user and does not require an authorization check; an automated technique that detects the absence of authorization may report false positives.

Effectiveness: Limited

Automated Dynamic Analysis

Automated dynamic analysis may find many or all possible interfaces that do not require authorization, but manual analysis is required to determine if the lack of authorization violates business logic

Manual Analysis

This weakness can be detected using tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session.

Specifically, manual static analysis is useful for evaluating the correctness of custom authorization mechanisms.

Effectiveness: Moderate

These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules. However, manual efforts might not achieve desired code coverage within limited time constraints.

Demonstrative Examples

Example 1

The following program could be part of a bulletin board system that allows users to send private messages to each other. This program intends to authenticate the user before deciding whether a private message should be displayed. Assume that LookupMessageObject() ensures that the $id argument is numeric, constructs a filename based on that id, and reads the message details from that file. Also assume that the program stores all private messages for all users in the same directory.

(Bad Code)

Example Language: **Perl**

```
sub DisplayPrivateMessage {
my($id) = @_;
my $Message = LookupMessageObject($id);
print "From: " . encodeHTML($Message->{from}) . "<br>\n";
print "Subject: " . encodeHTML($Message->{subject}) . "\n";
print "<hr>\n";
print "Body: " . encodeHTML($Message->{body}) . "\n";
}

my $q = new CGI;
# For purposes of this example, assume that CWE-309 and
# CWE-523 do not apply.
if (! AuthenticateUser($q->param('username'), $q->param('password'))) {
ExitError("invalid username or password");
}

my $id = $q->param('id');
DisplayPrivateMessage($id);
```

While the program properly exits if authentication fails, it does not ensure that the message is addressed to the user. As a result, an authenticated attacker could provide any arbitrary identifier and read private messages that were intended for other users.

One way to avoid this problem would be to ensure that the "to" field in the message object matches the username of the authenticated user.

Observed Examples

Reference	Description
CVE-2009-3168	Web application does not restrict access to admin scripts, allowing authenticated users to reset administrative passwords.

CVE-2009-2960	Web application does not restrict access to admin scripts, allowing authenticated users to modify passwords of other users.
CVE-2009-3597	Web application stores database file under the web root with insufficient access control (CWE-219), allowing direct request.
CVE-2009-2282	Terminal server does not check authorization for guest access.
CVE-2009-3230	Database server does not use appropriate privileges for certain sensitive operations.
CVE-2009-2213	Gateway uses default "Allow" configuration for its authorization settings.
CVE-2009-0034	Chain: product does not properly interpret a configuration option for a system group, allowing users to gain privileges.
CVE-2008-6123	Chain: SNMP product does not properly parse a configuration option for which hosts are allowed to connect, allowing unauthorized IP addresses to connect.
CVE-2008-5027	System monitoring software allows users to bypass authorization by creating custom forms.
CVE-2008-7109	Chain: reliance on client-side security (CWE-602) allows attackers to bypass authorization using a custom client.
CVE-2008-3424	Chain: product does not properly handle wildcards in an authorization policy list, allowing unintended access.
CVE-2009-3781	Content management system does not check access permissions for private files, allowing others to view those files.
CVE-2008-4577	ACL-based protection mechanism treats negative access rights as if they are positive, allowing bypass of intended restrictions.
CVE-2008-6548	Product does not check the ACL of a page accessed using an "include" directive, allowing attackers to read unauthorized files.
CVE-2007-2925	Default ACL list for a DNS server does not set certain ACLs, allowing unauthorized DNS queries.
CVE-2006-6679	Product relies on the X-Forwarded-For HTTP header for authorization, allowing unintended access by spoofing the header.
CVE-2005-3623	OS kernel does not check for a certain privilege before setting ACLs for files.
CVE-2005-2801	Chain: file-system code performs an incorrect comparison (CWE-697), preventing defauls ACLs from being properly applied.
CVE-2001-1155	Chain: product does not properly check the result of a reverse DNS lookup because of operator precedence (CWE-783), allowing bypass of DNS-based access restrictions.

Potential Mitigations

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully mapping roles with data and functionality. Use role-based access control (RBAC) to enforce the roles at the appropriate boundaries.

Note that this approach may not protect against horizontal authorization, i.e., it will not protect a user from attacking others with the same role.

Phase: Architecture and Design

Ensure that you perform access control checks related to your business logic. These checks may be different than the access control checks that you apply to more generic resources such as files, connections, processes, memory, and database records. For example, a database may restrict access for medical records to a specific database user, but each record might only be intended to be accessible to the patient and the patient's doctor.

Phase: Architecture and Design

Strategy: Libraries or Frameworks

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

For example, consider using authorization frameworks such as the JAAS Authorization Framework and the OWASP ESAPI Access Control feature.

Phase: Architecture and Design

For web applications, make sure that the access control mechanism is enforced correctly at the server side on every page. Users should not be able to access any unauthorized functionality or information by simply requesting direct access to that page.

One way to do this is to ensure that all pages containing sensitive information are not cached, and that all such pages restrict access to requests that are accompanied by an active and authenticated session token associated with a user who has the required permissions to access that page.

Phases: System Configuration; Installation

Use the access control capabilities of your operating system and server environment and define your access control lists accordingly. Use a "default deny" policy when defining these ACLs.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	254	Security Features	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Weakness Class	284	Access Control (Authorization) Issues	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ChildOf	Category	721	OWASP Top Ten 2007 Category A10 - Failure to Restrict URL Access	**Weaknesses in OWASP Top Ten (2007) (primary)629**
ChildOf	Category	723	OWASP Top Ten 2004 Category A2 - Broken Access Control	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
ParentOf	Weakness Variant	219	Sensitive Data Under Web Root	**Research Concepts (primary)1000**
ParentOf	Weakness Base	551	Incorrect Behavior Order: Authorization Before Parsing and Canonicalization	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Class	638	Failure to Use Complete Mediation	Research Concepts1000
ParentOf	Weakness Base	804	Guessable CAPTCHA	**Development Concepts (primary)699** **Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Missing Access Control
OWASP Top Ten 2007	A10	CWE More Specific	Failure to Restrict URL Access
OWASP Top Ten 2004	A2	CWE More Specific	Broken Access Control

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
1	Accessing Functionality Not Properly Constrained by ACLs	
13	Subverting Environment Variable Values	

17	Accessing, Modifying or Executing Executable Files
87	Forceful Browsing
39	Manipulating Opaque Client-based Data Tokens
45	Buffer Overflow via Symbolic Links
51	Poison Web Service Registry
59	Session Credential Falsification through Prediction
60	Reusing Session IDs (aka Session Replay)
77	Manipulating User-Controlled Variables
76	Manipulating Input to File System Calls
104	Cross Zone Scripting

References

NIST. "Role Based Access Control and Role Based Security". <http://csrc.nist.gov/groups/SNS/rbac/>.

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 4, "Authorization" Page 114; Chapter 6, "Determining Appropriate Access Control" Page 171. 2nd Edition. Microsoft. 2002.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	7 Pernicious Kingdoms		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Other Notes, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences, Description, Likelihood of Exploit, Name, Other Notes, Potential Mitigations, References, Relationships		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Description, Related Attack Patterns		
2009-07-27	CWE Content Team	MITRE	Internal
	updated Relationships		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Type		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Relationships		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Detection Factors, Potential Mitigations, References, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		

Previous Entry Names

Change Date	Previous Entry Name
2009-01-12	Missing or Inconsistent Access Control

BACK TO TOP

Incorrect Permission Assignment for Critical Resource

Weakness ID: 732 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The software specifies permissions for a security-critical resource in a way that allows that resource to be read or modified by unintended actors.

Extended Description

When a resource is given a permissions setting that provides access to a wider range of actors than required, it could lead to the disclosure of sensitive information, or the modification of that resource by unintended parties. This is especially dangerous when the resource is related to program configuration, execution or sensitive user data.

Time of Introduction

- Architecture and Design
- Implementation
- Installation
- Operation

Applicable Platforms

Languages

Language-independent

Modes of Introduction

The developer may set loose permissions in order to minimize problems when the user first runs the program, then create documentation stating that permissions should be tightened. Since system administrators and users do not always read the documentation, this can result in insecure permissions being left unchanged.

The developer might make certain assumptions about the environment in which the software runs - e.g., that the software is running on a single-user system, or the software is only accessible to trusted administrators. When the software is running in a different environment, the permissions become a problem.

Common Consequences

Scope	Effect
Confidentiality	An attacker may be able to read sensitive information from the associated resource, such as credentials or configuration information stored in a file.
Integrity	An attacker may be able to modify critical properties of the associated resource to gain privileges, such as replacing a world-writable executable with a Trojan horse.
Availability	An attacker may be able to destroy or corrupt critical data in the associated resource, such as deletion of records from a database.

Likelihood of Exploit

Medium to High

Detection Methods

Automated Static Analysis

Automated static analysis may be effective in detecting permission problems for system resources such as files, directories, shared memory, device interfaces, etc. Automated techniques may be able to detect the use of library functions that modify permissions, then analyze function calls for arguments that contain potentially insecure values.

However, since the software's intended security policy might allow loose permissions for certain operations (such as publishing a file on a web server), automated static analysis may produce some false positives - i.e., warnings that do not have any security consequences or require any code changes.

When custom permissions models are used - such as defining who can read messages in a particular forum in a bulletin board system - these can be difficult to detect using automated static analysis. It may be possible to define custom signatures that identify any custom functions that implement the permission checks and assignments.

Automated Dynamic Analysis

Automated dynamic analysis may be effective in detecting permission problems for system resources such as files, directories, shared memory, device interfaces, etc.

However, since the software's intended security policy might allow loose permissions for certain operations (such as publishing a file on a web server), automated dynamic analysis may produce some false positives - i.e., warnings that do not have any security consequences or require any code changes.

When custom permissions models are used - such as defining who can read messages in a particular forum in a bulletin board system - these can be difficult to detect using automated dynamic analysis. It may be possible to define custom signatures that identify any custom functions that implement the permission checks and assignments.

Manual Static Analysis

Manual static analysis may be effective in detecting the use of custom permissions models and functions. The code could then be examined to identifying usage of the related functions. Then the human analyst could evaluate permission assignments in the context of the intended security model of the software.

Manual Dynamic Analysis

Manual dynamic analysis may be effective in detecting the use of custom permissions models and functions. The program could then be executed with a focus on exercising code paths that are related to the custom permissions. Then the human analyst could evaluate permission assignments in the context of the intended security model of the software.

Fuzzing

Fuzzing is not effective in detecting this weakness.

Demonstrative Examples

Example 1

The following code sets the umask of the process to 0 before creating a file and writing "Hello world" into the file.

(Bad Code)

Example Language: **C**

```
#define OUTFILE "hello.out"

umask(0);
FILE *out;
/* Ignore CWE-59 (link following) for brevity */
out = fopen(OUTFILE, "w");
if (out) {
fprintf(out, "hello world!\n");
fclose(out);
}
```

After running this program on a UNIX system, running the "ls -l" command might return the following output:

(Result)

```
-rw-rw-rw- 1 username 13 Nov 24 17:58 hello.out
```

The "rw-rw-rw-" string indicates that the owner, group, and world (all users) can read the file and write to it.

Example 2

The following code snippet might be used as a monitor to periodically record whether a web site is alive. To ensure that the file can always be modified, the code uses chmod() to make the file world-writable.

(Bad Code)

Example Language: **Perl**

```
$fileName = "secretFile.out";

if (-e $fileName) {
chmod 0777, $fileName;
}
```

```
my $outFH;
if (! open($outFH, ">>$fileName")) {
ExitError("Couldn't append to $fileName: $!");
}
my $dateString = FormatCurrentTime();
my $status = IsHostAlive("cwe.mitre.org");
print $outFH "$dateString cwe status: $status!\n";
close($outFH);
```

The first time the program runs, it might create a new file that inherits the permissions from its environment. A file listing might look like:

(Result)

```
-rw-r--r-- 1 username 13 Nov 24 17:58 secretFile.out
```

This listing might occur when the user has a default umask of 022, which is a common setting. Depending on the nature of the file, the user might not have intended to make it readable by everyone on the system.

The next time the program runs, however - and all subsequent executions - the chmod will set the file's permissions so that the owner, group, and world (all users) can read the file and write to it:

(Result)

```
-rw-rw-rw- 1 username 13 Nov 24 17:58 secretFile.out
```

Perhaps the programmer tried to do this because a different process uses different permissions that might prevent the file from being updated.

Example 3

The following command recursively sets world-readable permissions for a directory and all of its children:

(Bad Code)

Example Language: **Shell**

```
chmod -R ugo+r DIRNAME
```

If this command is run from a program, the person calling the program might not expect that all the files under the directory will be world-readable. If the directory is expected to contain private data, this could become a security problem.

Observed Examples

Reference	Description
CVE-2009-3482	Anti-virus product sets insecure "Everyone: Full Control" permissions for files under the "Program Files" folder, allowing attackers to replace executables with Trojan horses.
CVE-2009-3897	Product creates directories with 0777 permissions at installation, allowing users to gain privileges and access a socket used for authentication.
CVE-2009-3489	Photo editor installs a service with an insecure security descriptor, allowing users to stop or start the service, or execute commands as SYSTEM.
CVE-2009-3289	Library function copies a file to a new target and uses the source file's permissions for the target, which is incorrect when the source file is a symbolic link, which typically has 0777 permissions.
CVE-2009-0115	Device driver uses world-writable permissions for a socket file, allowing attackers to inject arbitrary commands.
CVE-2009-1073	LDAP server stores a cleartext password in a world-readable file.
CVE-2009-0141	Terminal emulator creates TTY devices with world-writable permissions, allowing an attacker to write to the terminals of other users.

CVE-2008-0662	VPN product stores user credentials in a registry key with "Everyone: Full Control" permissions, allowing attackers to steal the credentials.
CVE-2008-0322	Driver installs its device interface with "Everyone: Write" permissions.
CVE-2009-3939	Driver installs a file with world-writable permissions.
CVE-2009-3611	Product changes permissions to 0777 before deleting a backup; the permissions stay insecure for subsequent backups.
CVE-2007-6033	Product creates a share with "Everyone: Full Control" permissions, allowing arbitrary program execution.
CVE-2007-5544	Product uses "Everyone: Full Control" permissions for memory-mapped files (shared memory) in inter-process communication, allowing attackers to tamper with a session.
CVE-2005-4868	Database product uses read/write permissions for everyone for its shared memory, allowing theft of credentials.
CVE-2004-1714	Security product uses "Everyone: Full Control" permissions for its configuration files.
CVE-2001-0006	"Everyone: Full Control" permissions assigned to a mutex allows users to disable network connectivity.
CVE-2002-0969	Chain: database product contains buffer overflow that is only reachable through a .ini configuration file - which has "Everyone: Full Control" permissions.

Potential Mitigations

Phase: Implementation

When using a critical resource such as a configuration file, check to see if the resource has insecure permissions (such as being modifiable by any regular user), and generate an error or even exit the software if there is a possibility that the resource could have been modified by an unauthorized party.

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully defining distinct user groups, privileges, and/or roles. Map these against data, functionality, and the related resources. Then set the permissions accordingly. This will allow you to maintain more fine-grained control over your resources.

Phases: Implementation; Installation

During program startup, explicitly set the default permissions or umask to the most restrictive setting possible. Also set the appropriate permissions during program installation. This will prevent you from inheriting insecure permissions from any user who installs or runs the program.

Phase: System Configuration

For all configuration files, executables, and libraries, make sure that they are only readable and writable by the software's administrator.

Phase: Documentation

Do not suggest insecure configuration changes in your documentation, especially if those configurations can extend to resources and other software that are outside the scope of your own software.

Phase: Installation

Do not assume that the system administrator will manually change the configuration to the settings that you recommend in the manual.

Phase: Testing

Use tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session. These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules.

Phase: Testing

Use monitoring tools that examine the software's process as it interacts with the operating system and the network. This technique is useful in cases when source code is unavailable, if the software was not developed by you, or if you want to verify that the build phase did not introduce any new weaknesses. Examples include debuggers that directly attach to the running process; system-call tracing utilities such as truss (Solaris) and strace (Linux); system activity monitors such as FileMon, RegMon, Process Monitor, and other Sysinternals utilities (Windows); and sniffers and protocol analyzers that monitor network traffic.

Attach the monitor to the process and watch for library functions or system calls on OS resources such as files, directories, and shared memory. Examine the arguments to these calls to infer which permissions are being used.

Note that this technique is only useful for permissions issues related to system resources. It is not likely to detect application-level business rules that are related to permissions, such as if a user of a blog system marks a post as "private," but the blog system inadvertently marks it as "public."

Phases: Testing; System Configuration

Ensure that your software runs properly under the Federal Desktop Core Configuration (FDCC) or an equivalent hardening configuration guide, which many organizations use to limit the attack surface and potential risk of deployed software.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	275	Permission Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	668	Exposure of Resource to Wrong Sphere	**Research Concepts (primary)1000**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
RequiredBy	Compound Element: Composite	689	Permission Race Condition During Resource Copy	Research Concepts1000
ParentOf	Weakness Variant	276	Incorrect Default Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	277	Insecure Inherited Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	278	Insecure Preserved Inherited Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	279	Incorrect Execution-Assigned Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Base	281	Improper Preservation of Permissions	**Research Concepts (primary)1000**

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
232	Exploitation of Privilege/Trust	
1	Accessing Functionality Not Properly Constrained by ACLs	
17	Accessing, Modifying or Executing Executable Files	
60	Reusing Session IDs (aka Session Replay)	
61	Session Fixation	
62	Cross Site Request Forgery (aka Session Riding)	
122	Exploitation of Authorization	
180	Exploiting Incorrectly Configured Access Control Security Levels	
234	Hijacking a privileged process	

References

Mark Dowd, John McDonald and Justin Schuh. "The Art of Software Security Assessment". Chapter 9, "File Permissions." Page 495.. 1st Edition. Addison Wesley. 2006.

John Viega and Gary McGraw. "Building Secure Software". Chapter 8, "Access Control." Page 194.. 1st Edition. Addison-Wesley. 2002.

Maintenance Notes

The relationships between privileges, permissions, and actors (e.g. users and groups) need further refinement within the Research view. One complication is that these concepts apply to two different pillars, related to control of resources (CWE-664) and protection mechanism failures (CWE-396).

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
2008-09-08			Internal CWE Team
	new weakness-focused entry for Research view.		

Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2009-01-12	CWE Content Team	MITRE	Internal
	updated Description, Likelihood of Exploit, Name, Potential Mitigations, Relationships		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Potential Mitigations, Related Attack Patterns		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Name		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Potential Mitigations, References		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Potential Mitigations, Related Attack Patterns		

Previous Entry Names	
Change Date	**Previous Entry Name**
2009-01-12	Insecure Permission Assignment for Resource
2009-05-27	Insecure Permission Assignment for Critical Resource

BACK TO TOP

Exposure of System Data to Unauthorized Control Sphere

Risk

What might happen

System data can provide attackers with valuable insights on systems and services they are targeting - any type of system data, from service version to operating system fingerprints, can assist attackers to hone their attack, correlate data with known vulnerabilities or focus efforts on developing new attacks against specific technologies.

Cause

How does it happen

System data is read and subsequently exposed where it might be read by untrusted entities.

General Recommendations

How to avoid it

Consider the implications of exposure of the specified input, and expected level of access to the specified output. If not required, consider removing this code, or modifying exposed information to exclude potentially sensitive system data.

Source Code Examples

Java

Leaking Environment Variables in JSP Web-Page

```
String envVarValue = System.getenv(envVar);
if (envVarValue == null) {
    out.println("Environment variable is not defined:");
    out.println(System.getenv());
} else {
    //[..]
};
```

TOCTOU

Risk

What might happen

At best, a Race Condition may cause errors in accuracy, overidden values or unexpected behavior that may result in denial-of-service. At worst, it may allow attackers to retrieve data or bypass security processes by replaying a controllable Race Condition until it plays out in their favor.

Cause

How does it happen

Race Conditions occur when a public, single instance of a resource is used by multiple concurrent logical processes. If the these logical processes attempt to retrieve and update the resource without a timely management system, such as a lock, a Race Condition will occur.

An example for when a Race Condition occurs is a resource that may return a certain value to a process for further editing, and then updated by a second process, resulting in the original process' data no longer being valid. Once the original process edits and updates the incorrect value back into the resource, the second process' update has been overwritten and lost.

General Recommendations

How to avoid it

When sharing resources between concurrent processes across the application ensure that these resources are either thread-safe, or implement a locking mechanism to ensure expected concurrent activity.

Source Code Examples

Java

Different Threads Increment and Decrement The Same Counter Repeatedly, Resulting in a Race Condition

```
     public static int counter = 0;
     public static void start() throws InterruptedException {
            incrementCounter ic;
            decrementCounter dc;
            while(counter == 0) {
                   counter = 0;
                   ic = new incrementCounter();
                   dc = new decrementCounter();
                   ic.start();
                   dc.start();
                   ic.join();
                   dc.join();
            }
            System.out.println(counter); //Will stop and return either -1 or 1 due to race
condition over counter
      }

     public static class incrementCounter extends Thread {
         public void run() {
            counter++;
         }
```

```
        }

        public static class decrementCounter extends Thread {
            public void run() {
               counter--;
            }
        }
```

Different Threads Increment and Decrement The Same Thread-Safe Counter Repeatedly, Never Resulting in a Race Condition

```
        public static int counter = 0;
        public static Object lock = new Object();

        public static void start() throws InterruptedException {
               incrementCounter ic;
               decrementCounter dc;
               while(counter == 0) { // because of proper locking, this condition is never false
                      counter = 0;
                      ic = new incrementCounter();
                      dc = new decrementCounter();
                      ic.start();
                      dc.start();
                      ic.join();
                      dc.join();
               }
               System.out.println(counter); // Never reached
        }

        public static class incrementCounter extends Thread {
            public void run() {
               synchronized (lock) {
                      counter++;
               }
            }
        }

        public static class decrementCounter extends Thread {
            public void run() {
               synchronized (lock) {
                      counter--;
               }
            }
        }
```

Use of sizeof() on a Pointer Type

Weakness ID: 467 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Integrity	This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(Bad Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(Good Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(Bad Code)

```
/* Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */

char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```

```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strncmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strncmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In AuthenticateUser(), because sizeof() is applied to a parameter with an array type, the sizeof() call might return 4 on many modern architectures. As a result, the strncmp() call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "sizeof(*pointer)" instead of "sizeof(pointer)", unless you intend to run sizeof() on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of sizeof() on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of sizeof(pointer) indicates a bug.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	465	Pointer Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	682	Incorrect Calculation	**Research Concepts (primary)1000**
ChildOf	Category	737	CERT C Secure Coding Section 03 - Expressions (EXP)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
CanPrecede	Weakness Base	131	Incorrect Calculation of Buffer Size	Research Concepts1000

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Use of sizeof() on a pointer type
CERT C Secure Coding	ARR01-C		Do not apply the sizeof operator to a pointer when taking the size of an array
CERT C Secure Coding	EXP01-C		Do not take the size of a pointer to determine the size of the pointed-to type

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator
2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type". <https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci updated Time of Introduction	Cigital	External
2008-08-01	 added/updated white box definitions	KDM Analytics	External
2008-09-08	CWE Content Team updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities	MITRE	Internal
2008-11-24	CWE Content Team updated Relationships, Taxonomy Mappings	MITRE	Internal
2009-03-10	CWE Content Team updated Demonstrative Examples	MITRE	Internal
2009-12-28	CWE Content Team updated Demonstrative Examples	MITRE	Internal
2010-02-16	CWE Content Team updated Relationships	MITRE	Internal

BACK TO TOP

Improper Validation of Array Index

Weakness ID: 129 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The product uses untrusted input when calculating or using an array index, but the product does not validate or incorrectly validates the index to ensure the index references a valid position within the array.

Alternate Terms

out-of-bounds array index

index-out-of-range

array index underflow

Time of Introduction

- Implementation

Applicable Platforms

Languages

C: *(Often)*

C++: *(Often)*

Language-independent

Common Consequences

Scope	Effect
Integrity Availability	Unchecked array indexing will very likely result in the corruption of relevant memory and perhaps instructions, leading to a crash, if the values are outside of the valid memory area.
Integrity	If the memory corrupted is data, rather than instructions, the system will continue to function with improper values.
Confidentiality Integrity	Unchecked array indexing can also trigger out-of-bounds read or write operations, or operations on the wrong objects; i.e., "buffer overflows" are not always the result. This may result in the exposure or modification of sensitive data.
Integrity	If the memory accessible by the attacker can be effectively controlled, it may be possible to execute arbitrary code, as with a standard buffer overflow and possibly without the use of large inputs if a precise index can be controlled.
Integrity Availability Confidentiality	A single fault could allow either an overflow (CWE-788) or underflow (CWE-786) of the array index. What happens next will depend on the type of operation being performed out of bounds, but can expose sensitive information, cause a system crash, or possibly lead to arbitrary code execution.

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

This weakness can often be detected using automated static analysis tools. Many modern tools use data flow analysis or constraint-based techniques to minimize the number of false positives.

Automated static analysis generally does not account for environmental considerations when reporting out-of-bounds memory operations. This can make it difficult for users to determine which warnings should be investigated first. For example, an analysis tool might report array index errors that originate from command line arguments in a program that is not expected to run with setuid or other special privileges.

Effectiveness: High

This is not a perfect solution, since 100% accuracy and coverage are not feasible.

Automated Dynamic Analysis

This weakness can be detected using dynamic tools and techniques that interact with the software using large test suites with many diverse inputs, such as fuzz testing (fuzzing), robustness testing, and fault injection. The software's operation may slow down, but it should not become unstable, crash, or generate incorrect results.

Black Box

Black box methods might not get the needed code coverage within limited time constraints, and a dynamic test might not produce any noticeable side effects even if it is successful.

Demonstrative Examples

Example 1

The following C/C++ example retrieves the sizes of messages for a pop3 mail server. The message sizes are retrieved from a socket that returns in a buffer the message number and the message size, the message number (num) and size (size) are extracted from the buffer and the message size is placed into an array using the message number for the array index.

(Bad Code)

Example Language: **C**

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
...
char buf[BUFFER_SIZE];
int ok;
int num, size;

// read values from socket and added to sizes array
while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
{

// continue read from socket until buf only contains '.'
if (DOTLINE(buf))
break;
else if (sscanf(buf, "%d %d", &num, &size) == 2)
sizes[num - 1] = size;
}
...
}
```

In this example the message number retrieved from the buffer could be a value that is outside the allowable range of indices for the array and could possibly be a negative number. Without proper validation of the value to be used for the array index an array overflow could occur and could potentially lead to unauthorized access to memory addresses and system crashes. The value of the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: **C**

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
...
char buf[BUFFER_SIZE];
int ok;
int num, size;

// read values from socket and added to sizes array
while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
{

// continue read from socket until buf only contains '.'
if (DOTLINE(buf))
break;
else if (sscanf(buf, "%d %d", &num, &size) == 2) {
if (num > 0 && num <= (unsigned)count)
sizes[num - 1] = size;
else
/* warn about possible attempt to induce buffer overflow */
report(stderr, "Warning: ignoring bogus data for message sizes returned by server.\n");
}
}
...
}
```

Example 2

In the code snippet below, an unchecked integer value is used to reference an object in an array.

(Bad Code)

Example Language: **Java**

```
public String getValue(int index) {
return array[index];
}
```

If index is outside of the range of the array, this may result in an ArrayIndexOutOfBounds Exception being raised.

Example 3

In the following Java example the method displayProductSummary is called from a Web service servlet to retrieve product summary information for display to the user. The servlet obtains the integer value of the product number from the user and passes it to the displayProductSummary method. The displayProductSummary method passes the integer value of the product number to the getProductSummary method which obtains the product summary from the array object containing the project summaries using the integer value of the product number as the array index.

(Bad Code)

Example Language: **Java**

```
// Method called from servlet to obtain product information
public String displayProductSummary(int index) {

String productSummary = new String("");

try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
return products[index];
}
```

In this example the integer value used as the array index that is provided by the user may be outside the allowable range of indices for the array which may provide unexpected results or may comes the application to fail. The integer value used for the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: **Java**

```
// Method called from servlet to obtain product information
public String displayProductSummary(int index) {

String productSummary = new String("");
try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
String productSummary = "";

if ((index >= 0) && (index < MAX_PRODUCTS)) {
productSummary = products[index];
}
else {
System.err.println("index is out of bounds");
throw new IndexOutOfBoundsException();
}

return productSummary;
}
```

An alternative in Java would be to use one of the collection objects such as ArrayList that will automatically generate an exception if an attempt is made to access an array index that is out of bounds.

(Good Code)

Example Language: **Java**

```
ArrayList productArray = new ArrayList(MAX_PRODUCTS);
...
try {
productSummary = (String) productArray.get(index);
} catch (IndexOutOfBoundsException ex) {...}
```

Observed Examples

Reference	Description
CVE-2005-0369	large ID in packet used as array index
CVE-2001-1009	negative array index as argument to POP LIST command
CVE-2003-0721	Integer signedness error leads to negative array index
CVE-2004-1189	product does not properly track a count and a maximum number, which can lead to resultant array index overflow.
CVE-2007-5756	chain: device driver for packet-capturing software allows access to an unintended IOCTL with resultant array index error.

Potential Mitigations

Phase: Architecture and Design

Strategies: Input Validation; Libraries or Frameworks

Use an input validation framework such as Struts or the OWASP ESAPI Validation API. If you use Struts, be mindful of weaknesses covered by the CWE-101 category.

Phase: Architecture and Design

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.

Even though client-side checks provide minimal benefits with respect to server-side security, they are still useful. First, they can support intrusion detection. If the server receives input that should have been rejected by the client, then it may be an indication of an attack. Second, client-side error-checking can provide helpful feedback to the user about the expectations for valid input. Third, there may be a reduction in server-side processing time for accidental input errors, although this is typically a small savings.

Phase: Requirements

Strategy: Language Selection

Use a language with features that can automatically mitigate or eliminate out-of-bounds indexing errors.

For example, Ada allows the programmer to constrain the values of a variable and languages such as Java and Ruby will allow the programmer to handle exceptions when an out-of-bounds index is accessed.

Phase: Implementation

Strategy: Input Validation

Assume all input is malicious. Use an "accept known good" input validation strategy (i.e., use a whitelist). Reject any input that does not strictly conform to specifications, or transform it into something that does. Use a blacklist to reject any unexpected inputs and detect potential attacks.

When accessing a user-controlled array index, use a stringent range of values that are within the target array. Make sure that you do not allow negative values to be used. That is, verify the minimum as well as the maximum of the range of acceptable values.

Phase: Implementation

Be especially careful to validate your input when you invoke code that crosses language boundaries, such as from an interpreted language to native code. This could create an unexpected interaction between the language boundaries. Ensure that you are not violating any of the expectations of the language with which you are interfacing. For example, even though Java may not be susceptible to buffer overflows, providing a large argument in a call to native code might trigger an overflow.

Weakness Ordinalities

Ordinality	Description
Resultant	The most common condition situation leading to unchecked array indexing is the use of loop index variables as buffer indexes. If the end condition for the loop is subject to a flaw, the index can grow or shrink unbounded, therefore causing a buffer overflow or underflow. Another common situation leading to this condition is the use of a function's return value, or the resulting value of a calculation directly as an index in to a buffer.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	20	Improper Input Validation	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ChildOf	Category	189	Numeric Errors	Development Concepts699
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	738	CERT C Secure Coding Section 04 - Integers (INT)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
ChildOf	Category	802	2010 Top 25 - Risky Resource Management	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
CanPrecede	Weakness Class	119	Failure to Constrain Operations within the Bounds of a Memory Buffer	Research Concepts1000
CanPrecede	Weakness Variant	789	Uncontrolled Memory Allocation	Research Concepts1000
PeerOf	Weakness Base	124	Buffer Underwrite ('Buffer Underflow')	Research Concepts1000

Theoretical Notes

An improperly validated array index might lead directly to the always-incorrect behavior of "access of array using out-of-bounds index."

Affected Resources

- Memory

f Causal Nature

Explicit

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Unchecked array indexing
PLOVER			INDEX - Array index overflow
CERT C Secure Coding	ARR00-C		Understand how arrays work
CERT C Secure Coding	ARR30-C		Guarantee that array indices are within the valid range
CERT C Secure Coding	ARR38-C		Do not add or subtract an integer to a pointer if the resulting value does not refer to a valid array element
CERT C Secure Coding	INT32-C		Ensure that operations on signed integers do not result in overflow

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
100	Overflow Buffers	

References

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 5, "Array Indexing Errors" Page 144. 2nd Edition. Microsoft. 2002.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Sean Eidemiller	Cigital	External
	added/updated demonstrative examples		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Description, Name, Relationships		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Observed Examples, Other Notes, Potential Mitigations, Theoretical Notes, Weakness Ordinalities		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Demonstrative Examples, Detection Factors, Likelihood of Exploit, Potential Mitigations, References, Related Attack Patterns, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Related Attack Patterns		

Previous Entry Names

Change Date	Previous Entry Name
2009-10-29	Unchecked Array Indexing

BACK TO TOP

Scanned Languages

Language	Hash Number	Change Date
CPP	4541647240435660	6/19/2024
Common	0105849645654507	6/19/2024