proxmark3 Scan Report

Project Name	proxmark3
Scan Start	Friday, June 21, 2024 10:48:53 PM
Preset	Checkmarx Default
Scan Time	00h:03m:50s
Lines Of Code Scanned	20242
Files Scanned	21
Report Creation Time	Friday, June 21, 2024 10:56:37 PM
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	3/100 (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

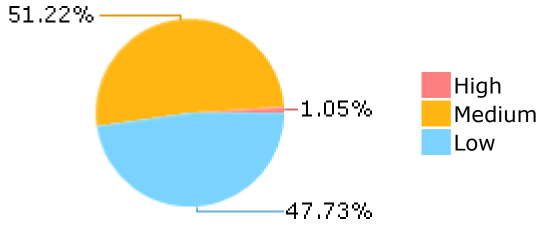
Results Limit

Results limit per query was set to 50

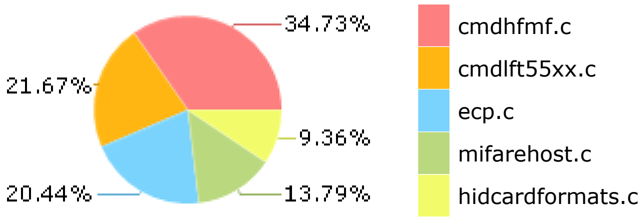
Selected Queries

Selected queries are listed in Result Summary

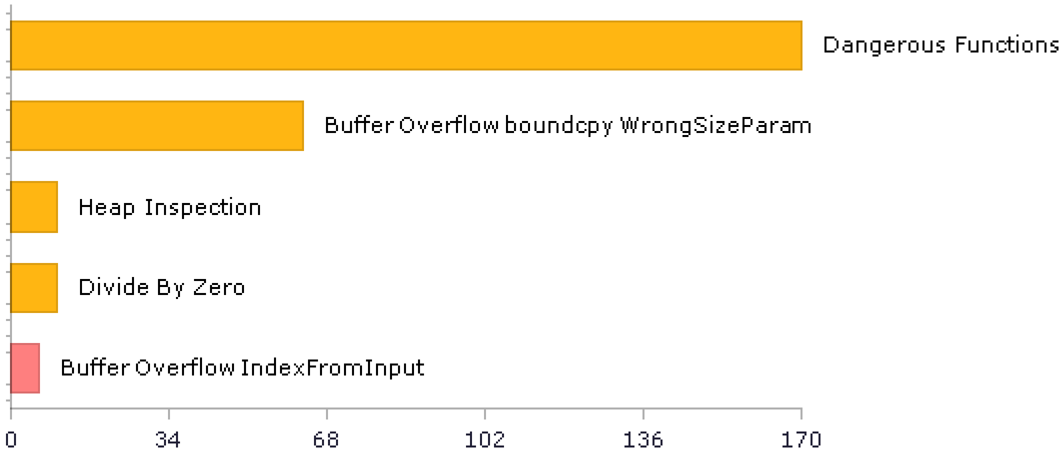
Result Summary



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	175	86
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	68	68
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	10	10
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	170	170
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	10	10
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	170	170
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	7	7
PCI DSS (3.2) - 6.5.2 - Buffer overflows	75	75
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.	20	20
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.	3	3
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.	0	0
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.	50	50
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.	10	10
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.	5	5

***** 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)	68	68
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)	0	0
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	0	0
SC-23 Session Authenticity (P1)*****	0	0
SC-28 Protection of Information at Rest (P1)	2	2
SC-4 Information in Shared Resources (P1)	10	10
SC-5 Denial of Service Protection (P1)*****	104	20
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*****	23	19
SI-11 Error Handling (P2)*****	68	68
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	9	8

***** 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

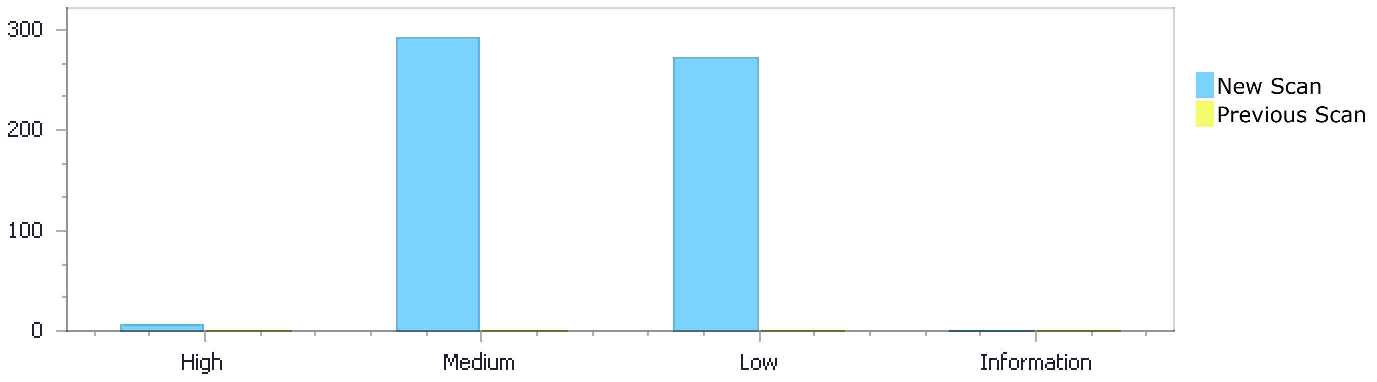
Category	Issues Found	Best Fix Locations
Must audit	0	0
Check	0	0
Optional	0	0

Results Distribution By Status

First scan of the project

	High	Medium	Low	Information	Total
New Issues	6	293	273	0	572
Recurrent Issues	0	0	0	0	0
Total	6	293	273	0	572

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	6	293	273	0	572
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	6	293	273	0	572

Result Summary

Vulnerability Type	Occurrences	Severity
Buffer Overflow IndexFromInput	6	High
Dangerous Functions	170	Medium
Buffer Overflow boundcpy WrongSizeParam	63	Medium
Divide By Zero	10	Medium
Heap Inspection	10	Medium

Stored Buffer Overflow boundcpy	8	Medium
Buffer Overflow AddressOfLocalVarReturned	7	Medium
MemoryFree on StackVariable	6	Medium
Integer Overflow	5	Medium
Memory Leak	5	Medium
Use of Zero Initialized Pointer	4	Medium
Double Free	2	Medium
Wrong Size t Allocation	2	Medium
Use of Uninitialized Variable	1	Medium
NULL Pointer Dereference	84	Low
Unchecked Return Value	68	Low
Improper Resource Access Authorization	48	Low
Incorrect Permission Assignment For Critical Resources	20	Low
TOCTOU	20	Low
Unchecked Array Index	10	Low
Potential Off by One Error in Loops	7	Low
Sizeof Pointer Argument	7	Low
Use of Sizeof On a Pointer Type	4	Low
Arithmenic Operation On Boolean	3	Low
Information Exposure Through Comments	2	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
proxmark3/cmdhfmf.c	78
proxmark3/mifarehost.c	43
proxmark3/hidcardformats.c	38
proxmark3/mifareutil.c	33
proxmark3/cmdlft55xx.c	27
proxmark3/lfdemod.c	21
proxmark3/lobject.c	14
proxmark3/hardnested_bruteforce.c	11
proxmark3/elite_crack.c	11
proxmark3/cmdlf.c	10

Scan Results Details

Buffer Overflow IndexFromInput

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

Categories

OWASP Top 10 2017: A1-Injection

Description

Buffer Overflow IndexFromInput\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=1
Status	New

The size of the buffer used by read_bench_data in i, at line 370 of proxmark3/hardnested_bruteforce.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that read_bench_data passes to BinaryExpr, at line 370 of proxmark3/hardnested_bruteforce.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	409	416
Object	BinaryExpr	i

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static bool read_bench_data(statelist_t *test_candidates) {

```
....
409.                bytes_read = fread(test_candidates->states[EVEN_STATE] + states_read, 1, sizeof(uint32_t), benchfile);
....
416.                test_candidates->states[EVEN_STATE][i] = test_candidates->states[EVEN_STATE][i-states_read];
```

Buffer Overflow IndexFromInput\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=2
Status	New

The size of the buffer used by read_bench_data in EVEN_STATE, at line 370 of proxmark3/hardnested_bruteforce.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that read_bench_data passes to BinaryExpr, at line 370 of proxmark3/hardnested_bruteforce.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	409	416
Object	BinaryExpr	EVEN_STATE

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static bool read_bench_data(statelist_t *test_candidates) {

```
....
409.                bytes_read = fread(test_candidates->states[EVEN_STATE]
+ states_read, 1, sizeof(uint32_t), benchfile);
....
416.                test_candidates->states[EVEN_STATE][i] =
test_candidates->states[EVEN_STATE][i-states_read];
```

Buffer Overflow IndexFromInput\Path 3:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=3
Status	New

The size of the buffer used by read_bench_data in i, at line 370 of proxmark3/hardnested_bruteforce.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that read_bench_data passes to BinaryExpr, at line 370 of proxmark3/hardnested_bruteforce.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	409	433
Object	BinaryExpr	i

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static bool read_bench_data(statelist_t *test_candidates) {

```
....
409.                bytes_read = fread(test_candidates->states[EVEN_STATE]
+ states_read, 1, sizeof(uint32_t), benchfile);
....
433.                test_candidates->states[ODD_STATE][i] =
test_candidates->states[ODD_STATE][i-states_read];
```

Buffer Overflow IndexFromInput\Path 4:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=4

Status	New

The size of the buffer used by read_bench_data in i, at line 370 of proxmark3/hardnested_bruteforce.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that read_bench_data passes to BinaryExpr, at line 370 of proxmark3/hardnested_bruteforce.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	426	433
Object	BinaryExpr	i

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static bool read_bench_data(statelist_t *test_candidates) {

```
....
426.                  bytes_read = fread(test_candidates->states[ODD_STATE]
+ states_read, 1, sizeof(uint32_t), benchfile);
....
433.                  test_candidates->states[ODD_STATE][i] =
test_candidates->states[ODD_STATE][i-states_read];
```

Buffer Overflow IndexFromInput\Path 5:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=5
Status	New

The size of the buffer used by read_bench_data in ODD_STATE, at line 370 of proxmark3/hardnested_bruteforce.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that read_bench_data passes to BinaryExpr, at line 370 of proxmark3/hardnested_bruteforce.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	409	433
Object	BinaryExpr	ODD_STATE

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static bool read_bench_data(statelist_t *test_candidates) {

```
....
409.                  bytes_read = fread(test_candidates->states[EVEN_STATE]
+ states_read, 1, sizeof(uint32_t), benchfile);
....
433.                  test_candidates->states[ODD_STATE][i] =
test_candidates->states[ODD_STATE][i-states_read];
```

Buffer Overflow IndexFromInput\Path 6:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=6
Status	New

The size of the buffer used by read_bench_data in ODD_STATE, at line 370 of proxmark3/hardnested_bruteforce.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that read_bench_data passes to BinaryExpr, at line 370 of proxmark3/hardnested_bruteforce.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	426	433
Object	BinaryExpr	ODD_STATE

Code Snippet

File Name proxmark3/hardnested_bruteforce.c

Method static bool read_bench_data(statelist_t *test_candidates) {

```
....
426.                bytes_read = fread(test_candidates->states[ODD_STATE]
+ states_read, 1, sizeof(uint32_t), benchfile);
....
433.                test_candidates->states[ODD_STATE][i] =
test_candidates->states[ODD_STATE][i-states_read];
```

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=1050058&projectid=50048&pathid=283
Status	New

The dangerous function, memcpy, was found in use at line 63 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	97	97

Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfWrBl(const char *Cmd)

```
....
97.    memcpy(c.d.asBytes, key, 6);
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=284
Status	New

The dangerous function, memcpy, was found in use at line 63 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	98	98
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfWrBl(const char *Cmd)

```
....
98.    memcpy(c.d.asBytes + 10, bldata, 16);
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=285
Status	New

The dangerous function, memcpy, was found in use at line 112 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	141	141
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfRdBl(const char *Cmd)

```
....
141.        memcpy(c.d.asBytes, key, 6);
```

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=286
Status	New

The dangerous function, memcpy, was found in use at line 174 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	208	208
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfRdSc(const char *Cmd)

```
....
208.        memcpy(c.d.asBytes, key, 6);
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=287
Status	New

The dangerous function, memcpy, was found in use at line 280 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	342	342
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfDump(const char *Cmd)

```
....
342.                        memcpy(c.d.asBytes, keys[0][sectorNo], 6);
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=288
Status	New

The dangerous function, memcpy, was found in use at line 280 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	378	378
Object	memcpy	memcpy

Code Snippet
File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfDump(const char *Cmd)

```
....
378.                                    memcpy(c.d.asBytes,
keys[0][sectorNo], 6);
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=289
Status	New

The dangerous function, memcpy, was found in use at line 280 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	385	385
Object	memcpy	memcpy

Code Snippet
File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfDump(const char *Cmd)

```
....
385.                                        memcpy(c.d.asBytes,
keys[k][sectorNo], 6);
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=290
Status	New

The dangerous function, memcpy, was found in use at line 280 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	396	396
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfDump(const char *Cmd)

```
....
396.                                        memcpy(c.d.asBytes,
keys[1][sectorNo], 6);
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=291
Status	New

The dangerous function, memcpy, was found in use at line 280 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	412	412
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfDump(const char *Cmd)

```
....
412.                                        memcpy(c.d.asBytes,
keys[0][sectorNo], 6);
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=292
Status	New

The dangerous function, memcpy, was found in use at line 280 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	427	427
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfDump(const char *Cmd)

```
....
427.                                  memcpy(data, keys[0][sectorNo], 6);
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=293
Status	New

The dangerous function, memcpy, was found in use at line 280 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	428	428
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfDump(const char *Cmd)

```
....
428.                                   memcpy(data + 10, keys[1][sectorNo],
6);
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=294
Status	New

The dangerous function, memcpy, was found in use at line 280 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	431	431
Object	memcpy	memcpy

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfDump(const char *Cmd)

```
....
431.
      memcpy(carddata[FirstBlockOfSector(sectorNo) + blockNo], data,
16);
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=295
Status	New

The dangerous function, memcpy, was found in use at line 460 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	526	526
Object	memcpy	memcpy

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfRestore(const char *Cmd)

```
....
526.                    memcpy(c.d.asBytes, key, 6);
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=296
Status	New

The dangerous function, memcpy, was found in use at line 460 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	552	552
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfRestore(const char *Cmd)

```
....
552.                    memcpy(c.d.asBytes + 10, bldata, 16);
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=297
Status	New

The dangerous function, memcpy, was found in use at line 1364 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1450	1450
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method void readerAttack(nonces_t ar_resp[], bool setEmulatorMem, bool doStandardAttack) {

```
....
1450.                         memcpy(c.d.asBytes, memBlock, 16);
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=298
Status	New

The dangerous function, memcpy, was found in use at line 1489 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1635	1635
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfSim(const char *Cmd) {

```
....
1635.                   memcpy(c.d.asBytes, uid, sizeof(uid));
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=299
Status	New

The dangerous function, memcpy, was found in use at line 1489 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1645	1645
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfSim(const char *Cmd) {

```
....
1645.                memcpy(ar_resp, resp.d.asBytes,
sizeof(ar_resp));
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=300
Status	New

The dangerous function, memcpy, was found in use at line 1489 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1670	1670
Object	memcpy	memcpy

Code Snippet	
File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfSim(const char *Cmd) {

```
....
1670.            memcpy(c.d.asBytes, uid, sizeof(uid));
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=301
Status	New

The dangerous function, memcpy, was found in use at line 1489 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1683	1683
Object	memcpy	memcpy

Code Snippet	
File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfSim(const char *Cmd) {

```
....
1683.                    memcpy(ar_resp, resp.d.asBytes,
sizeof(ar_resp));
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=302
Status	New

The dangerous function, memcpy, was found in use at line 2609 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2704	2704
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfSniff(const char *Cmd){

```
....
2704.                    memcpy(bufPtr, resp.d.asBytes, len);
```

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=303
Status	New

The dangerous function, memcpy, was found in use at line 2609 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2726	2726
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfSniff(const char *Cmd){

```
....
2726.                                              memcpy(uid, bufPtr + 2, 7);
```

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=304
Status	New

The dangerous function, memcpy, was found in use at line 2609 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2727	2727
Object	memcpy	memcpy

Code Snippet

File Name proxmark3/cmdhfmf.c

Method int CmdHF14AMfSniff(const char *Cmd){

```
....
2727.                                              memcpy(atqa, bufPtr + 2 + 7,
2);
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=305
Status	New

The dangerous function, memcpy, was found in use at line 2816 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2883	2883
Object	memcpy	memcpy

Code Snippet

File Name proxmark3/cmdhfmf.c

Method int CmdHF14AMfMAD(const char *cmd) {

```
....
2883.                memcpy(akey, g_mifare_ndef_key, 6);
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=306
Status	New

The dangerous function, memcpy, was found in use at line 2816 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2885	2885
Object	memcpy	memcpy

Code Snippet	
File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfMAD(const char *cmd) {

```
....
2885.                       memcpy(akey, key, 6);
```

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=307
Status	New

The dangerous function, memcpy, was found in use at line 2906 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2940	2940
Object	memcpy	memcpy

Code Snippet	
File Name	proxmark3/cmdhfmf.c
Method	int CmdHFMFNDEF(const char *cmd) {

```
....
2940.       memcpy(ndefkey, g_mifare_ndef_key, 6);
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=308
Status	New

The dangerous function, memcpy, was found in use at line 2906 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2942	2942
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHFMFNDEF(const char *cmd) {

```
....
2942.           memcpy(ndefkey, key, 6);
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=309
Status	New

The dangerous function, memcpy, was found in use at line 2906 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2987	2987
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHFMFNDEF(const char *cmd) {

```
....
2987.                   memcpy(&data[datalen], vsector, 16 * 3);
```

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=310
Status	New

The dangerous function, memcpy, was found in use at line 3010 in proxmark3/cmdhfmf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	3074	3074
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHFMFPersonalize(const char *cmd) {

```
....
3074.         memcpy(c.d.asBytes, key, 6);
```

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=311
Status	New

The dangerous function, memcpy, was found in use at line 249 in proxmark3/cmdlf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdlf.c	proxmark3/cmdlf.c
Line	329	329
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdlf.c
Method int CmdLFSetConfig(const char *Cmd)

```
....
329.        memcpy(c.d.asBytes,&config,sizeof(sample_config));
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=312
Status	New

The dangerous function, memcpy, was found in use at line 485 in proxmark3/cmdlf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdlf.c	proxmark3/cmdlf.c
Line	579	579
Object	memcpy	memcpy

Code Snippet

File Name	proxmark3/cmdlf.c
Method	int CmdLFfskSim(const char *Cmd)

```
....
579.        memcpy(c.d.asBytes, DemodBuffer, size);
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=313
Status	New

The dangerous function, memcpy, was found in use at line 587 in proxmark3/cmdlf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdlf.c	proxmark3/cmdlf.c
Line	673	673
Object	memcpy	memcpy

Code Snippet

File Name	proxmark3/cmdlf.c
Method	int CmdLFaskSim(const char *Cmd)

```
....
673.        memcpy(c.d.asBytes, DemodBuffer, size);
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=314
Status	New

The dangerous function, memcpy, was found in use at line 681 in proxmark3/cmdlf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdlf.c	proxmark3/cmdlf.c
Line	781	781
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdlf.c
Method int CmdLFpskSim(const char *Cmd)

```
....
781.        memcpy(c.d.asBytes, DemodBuffer, size);
```

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=315
Status	New

The dangerous function, memcpy, was found in use at line 1452 in proxmark3/cmdlft55xx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1489	1489
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method int CmdT55xxBruteForce(const char *Cmd) {

```
....
1489.            memcpy(filename, Cmd+2+cmd_offset, len);
```

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=316
Status	New

The dangerous function, memcpy, was found in use at line 168 in proxmark3/elite_crack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/elite_crack.c	proxmark3/elite_crack.c
Line	171	171
Object	memcpy	memcpy

Code Snippet

File Name	proxmark3/elite_crack.c
Method	void rk(uint8_t *key, uint8_t n, uint8_t *outp_key)

```
....
171.        memcpy(outp_key, key, 8);
```

Dangerous Functions\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=317
Status	New

The dangerous function, memcpy, was found in use at line 206 in proxmark3/elite_crack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/elite_crack.c	proxmark3/elite_crack.c
Line	258	258
Object	memcpy	memcpy

Code Snippet

File Name	proxmark3/elite_crack.c
Method	void hash2(uint8_t *key64, uint8_t *outp_keytable)

```
....
258.                    memcpy(outp_keytable+i*16,y[i],8);
```

Dangerous Functions\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=318
Status	New

The dangerous function, memcpy, was found in use at line 206 in proxmark3/elite_crack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/elite_crack.c	proxmark3/elite_crack.c
Line	259	259
Object	memcpy	memcpy

Code Snippet

File Name	proxmark3/elite_crack.c
Method	void hash2(uint8_t *key64, uint8_t *outp_keytable)

```
....
259.                    memcpy(outp_keytable+8+i*16,z[i],8);
```

Dangerous Functions\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=319
Status	New

The dangerous function, memcpy, was found in use at line 283 in proxmark3/elite_crack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/elite_crack.c	proxmark3/elite_crack.c
Line	287	287
Object	memcpy	memcpy

Code Snippet

File Name	proxmark3/elite_crack.c
Method	int _readFromDump(uint8_t dump[], dumpdata* item, uint8_t i)

```
....
287.        memcpy(item,dump+i*itemsize, itemsize);
```

Dangerous Functions\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=320
Status	New

The dangerous function, memcpy, was found in use at line 440 in proxmark3/elite_crack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/elite_crack.c	proxmark3/elite_crack.c
Line	452	452
Object	memcpy	memcpy

Code Snippet
File Name   proxmark3/elite_crack.c
Method   int calculateMasterKey(uint8_t first16bytes[], uint64_t master_key[])

```
....
452.        memcpy(y_0,first16bytes,8);
```

Dangerous Functions\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=321
Status	New

The dangerous function, memcpy, was found in use at line 440 in proxmark3/elite_crack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/elite_crack.c	proxmark3/elite_crack.c
Line	453	453
Object	memcpy	memcpy

Code Snippet
File Name   proxmark3/elite_crack.c
Method   int calculateMasterKey(uint8_t first16bytes[], uint64_t master_key[])

```
....
453.        memcpy(z_0,first16bytes+8,8);
```

Dangerous Functions\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=322
Status	New

The dangerous function, memcpy, was found in use at line 440 in proxmark3/elite_crack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/elite_crack.c	proxmark3/elite_crack.c
Line	482	482
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/elite_crack.c
Method int calculateMasterKey(uint8_t first16bytes[], uint64_t master_key[])

```
....
482.            memcpy(master_key, key64, 8);
```

Dangerous Functions\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=323
Status	New

The dangerous function, memcpy, was found in use at line 500 in proxmark3/elite_crack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/elite_crack.c	proxmark3/elite_crack.c
Line	511	511
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/elite_crack.c
Method int bruteforceDump(uint8_t dump[], size_t dumpsize, uint16_t keytable[])

```
....
511.                memcpy(attack,dump+i*itemsize, itemsize);
```

Dangerous Functions\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=324
Status	New

The dangerous function, memcpy, was found in use at line 252 in proxmark3/lobject.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/lobject.c	proxmark3/lobject.c
Line	256	256
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
256.          memcpy(out, source + 1, l * sizeof(char));
```

Dangerous Functions\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=325
Status	New

The dangerous function, memcpy, was found in use at line 252 in proxmark3/lobject.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/lobject.c	proxmark3/lobject.c
Line	264	264
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
264.        memcpy(out, source + 1, l * sizeof(char));
```

Dangerous Functions\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=326
Status	New

The dangerous function, memcpy, was found in use at line 252 in proxmark3/lobject.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/lobject.c	proxmark3/lobject.c
Line	268	268
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
268.        memcpy(out, source + 1 + l - bufflen, bufflen *
sizeof(char));
```

Dangerous Functions\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=327
Status	New

The dangerous function, memcpy, was found in use at line 252 in proxmark3/lobject.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/lobject.c	proxmark3/lobject.c
Line	284	284
Object	memcpy	memcpy

Code Snippet
File Name proxmark3/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
284.        memcpy(out, POS, (LL(POS) + 1) * sizeof(char));
```

Dangerous Functions\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=328
Status	New

The dangerous function, memcpy, was found in use at line 293 in proxmark3/lvm.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/lvm.c	proxmark3/lvm.c
Line	324	324
Object	memcpy	memcpy

Code Snippet

File Name	proxmark3/lvm.c
Method	void luaV_concat (lua_State *L, int total) {

```
....
324.            memcpy(buffer+tl, svalue(top-i), l * sizeof(char));
```

Dangerous Functions\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=329
Status	New

The dangerous function, memcpy, was found in use at line 1214 in proxmark3/mifarehost.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	1221	1221
Object	memcpy	memcpy

Code Snippet

File Name	proxmark3/mifarehost.c
Method	int DetectClassicPrng(void){

```
....
1221.       memcpy(c.d.asBytes, cmd, sizeof(cmd));
```

Dangerous Functions\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=330
Status	New

The dangerous function, memcpy, was found in use at line 224 in proxmark3/mifarehost.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	252	252
Object	memcpy	memcpy

Code Snippet

File Name proxmark3/mifarehost.c

Method static int mfCheckKeysEx(uint8_t blockNo, uint8_t keyType, uint16_t timeout14a, bool clear_trace, uint32_t keycnt, uint8_t *keys, uint64_t *found_key, bool fixed_nonce) {

```
....
252.            memcpy(c.d.asBytes, keys + i * bytes_per_key, max_keys
* bytes_per_key);
```

Dangerous Functions\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=331
Status	New

The dangerous function, memcpy, was found in use at line 295 in proxmark3/mifarehost.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	306	306
Object	memcpy	memcpy

Code Snippet

File Name proxmark3/mifarehost.c

Method	int mfCheckKeysSec(uint8_t sectorCnt, uint8_t keyType, uint16_t timeout14a, bool clear_trace, bool init, bool drop_field, uint8_t keycnt, uint8_t *keyBlock, sector_t *e_sector) {

```
....
306.        memcpy(c.d.asBytes, keyBlock, 6 * keycnt);
```

Dangerous Functions\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=332
Status	New

The dangerous function, memcpy, was found in use at line 505 in proxmark3/mifarehost.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	511	511
Object	memcpy	memcpy

Code Snippet	
File Name	proxmark3/mifarehost.c
Method	int mfnested(uint8_t blockNo, uint8_t keyType, uint16_t timeout14a, uint8_t *key, uint8_t trgBlockNo, uint8_t trgKeyType, uint8_t *resultKey, bool calibrate) {

```
....
511.        memcpy(c.d.asBytes, key, 6);
```

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=1050058&projectid=50048&pathid=103
Status	New

The size of the buffer used by CmdHF14AMfSim in uid, at line 1489 of proxmark3/cmdhfmf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that CmdHF14AMfSim passes to uid, at line 1489 of proxmark3/cmdhfmf.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1635	1635
Object	uid	uid

Code Snippet
File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfSim(const char *Cmd) {

```
....
1635.                           memcpy(c.d.asBytes, uid, sizeof(uid));
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=104
Status	New

The size of the buffer used by CmdHF14AMfSim in uid, at line 1489 of proxmark3/cmdhfmf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that CmdHF14AMfSim passes to uid, at line 1489 of proxmark3/cmdhfmf.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1670	1670
Object	uid	uid

Code Snippet
File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfSim(const char *Cmd) {

```
....
1670.                   memcpy(c.d.asBytes, uid, sizeof(uid));
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=105
Status	New

The size of the buffer used by CmdLFSetConfig in sample_config, at line 249 of proxmark3/cmdlf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that CmdLFSetConfig passes to sample_config, at line 249 of proxmark3/cmdlf.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/cmdlf.c	proxmark3/cmdlf.c
Line	329	329
Object	sample_config	sample_config

Code Snippet
File Name proxmark3/cmdlf.c
Method int CmdLFSetConfig(const char *Cmd)

```
....
329.        memcpy(c.d.asBytes,&config,sizeof(sample_config));
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=106
Status	New

The size of the buffer used by DetectClassicPrng in cmd, at line 1214 of proxmark3/mifarehost.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that DetectClassicPrng passes to cmd, at line 1214 of proxmark3/mifarehost.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	1221	1221
Object	cmd	cmd

Code Snippet
File Name proxmark3/mifarehost.c
Method int DetectClassicPrng(void){

```
....
1221.       memcpy(c.d.asBytes, cmd, sizeof(cmd));
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=107
Status	New

The size of the buffer used by mifare_sendcmd_short in dcmd, at line 97 of proxmark3/mifareutil.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mifare_sendcmd_short passes to dcmd, at line 97 of proxmark3/mifareutil.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/mifareutil.c	proxmark3/mifareutil.c
Line	105	105
Object	dcmd	dcmd

Code Snippet
File Name proxmark3/mifareutil.c
Method int mifare_sendcmd_short(struct Crypto1State *pcs, uint8_t crypted, uint8_t cmd, uint8_t data, uint8_t *answer, uint8_t *answer_parity, uint32_t *timing) {

```
....
105.        memcpy(ecmd, dcmd, sizeof(dcmd));
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=108
Status	New

The size of the buffer used by mbedtls_ecp_group_init in mbedtls_ecp_group, at line 293 of proxmark3/ecp.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mbedtls_ecp_group_init passes to mbedtls_ecp_group, at line 293 of proxmark3/ecp.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	298	298
Object	mbedtls_ecp_group	mbedtls_ecp_group

Code Snippet
File Name proxmark3/ecp.c
Method void mbedtls_ecp_group_init(mbedtls_ecp_group *grp)

```
....
298.      memset( grp, 0, sizeof( mbedtls_ecp_group ) );
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=109
Status	New

The size of the buffer used by HIDTryUnpack in hidproxcard_t, at line 588 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that HIDTryUnpack passes to hidproxcard_t, at line 588 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	595	595
Object	hidproxcard_t	hidproxcard_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool HIDTryUnpack(/* in */hidproxmessage_t* packed, /* in */bool ignoreParity){

```
....
595.    memset(&card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=110
Status	New

The size of the buffer used by Pack_H10301 in hidproxmessage_t, at line 21 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_H10301 passes to hidproxmessage_t, at line 21 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	22	22
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Pack_H10301(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
22.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=111
Status	New

The size of the buffer used by Unpack_H10301 in hidproxcard_t, at line 34 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_H10301 passes to hidproxcard_t, at line 34 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	35	35
Object	hidproxcard_t	hidproxcard_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Unpack_H10301(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
35.    memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=112
Status	New

The size of the buffer used by Pack_Tecom27 in hidproxmessage_t, at line 45 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_Tecom27 passes to hidproxmessage_t, at line 45 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	46	46
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Pack_Tecom27(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
46.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=113
Status	New

The size of the buffer used by Unpack_Tecom27 in hidproxcard_t, at line 56 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_Tecom27 passes to hidproxcard_t, at line 56 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	57	57
Object	hidproxcard_t	hidproxcard_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Unpack_Tecom27(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
57.    memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=114
Status	New

The size of the buffer used by Pack_2804W in hidproxmessage_t, at line 64 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_2804W passes to hidproxmessage_t, at line 64 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	65	65
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Pack_2804W(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
65.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=115

Status	New

The size of the buffer used by Unpack_2804W in hidproxcard_t, at line 84 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_2804W passes to hidproxcard_t, at line 84 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	85	85
Object	hidproxcard_t	hidproxcard_t

Code Snippet

File Name proxmark3/hidcardformats.c

Method bool Unpack_2804W(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
85.     memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=116
Status	New

The size of the buffer used by Pack_ATSW30 in hidproxmessage_t, at line 96 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_ATSW30 passes to hidproxmessage_t, at line 96 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	97	97
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet

File Name proxmark3/hidcardformats.c

Method bool Pack_ATSW30(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
97.     memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=500

	48&pathid=117
Status	New

The size of the buffer used by Unpack_ATSW30 in hidproxcard_t, at line 113 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_ATSW30 passes to hidproxcard_t, at line 113 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	114	114
Object	hidproxcard_t	hidproxcard_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Unpack_ATSW30(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
114.    memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=118
Status	New

The size of the buffer used by Pack_ADT31 in hidproxmessage_t, at line 123 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_ADT31 passes to hidproxmessage_t, at line 123 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	124	124
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Pack_ADT31(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
124.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 17:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=119
Status	New

The size of the buffer used by Unpack_ADT31 in hidproxcard_t, at line 135 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_ADT31 passes to hidproxcard_t, at line 135 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	136	136
Object	hidproxcard_t	hidproxcard_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Unpack_ADT31(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
136.    memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=120
Status	New

The size of the buffer used by Pack_Kastle in hidproxmessage_t, at line 143 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_Kastle passes to hidproxmessage_t, at line 143 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	144	144
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Pack_Kastle(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
144.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 19:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=121
Status	New

The size of the buffer used by Unpack_Kastle in hidproxcard_t, at line 158 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_Kastle passes to hidproxcard_t, at line 158 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	159	159
Object	hidproxcard_t	hidproxcard_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Unpack_Kastle(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
159.    memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=122
Status	New

The size of the buffer used by Pack_D10202 in hidproxmessage_t, at line 171 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_D10202 passes to hidproxmessage_t, at line 171 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	172	172
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Pack_D10202(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
172.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 21:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=123
Status	New

The size of the buffer used by Unpack_D10202 in hidproxcard_t, at line 184 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_D10202 passes to hidproxcard_t, at line 184 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	185	185
Object	hidproxcard_t	hidproxcard_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Unpack_D10202(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
185.    memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=124
Status	New

The size of the buffer used by Pack_H10306 in hidproxmessage_t, at line 195 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_H10306 passes to hidproxmessage_t, at line 195 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	196	196
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Pack_H10306(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
196.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=125
Status	New

The size of the buffer used by Unpack_H10306 in hidproxcard_t, at line 209 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_H10306 passes to hidproxcard_t, at line 209 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	210	210
Object	hidproxcard_t	hidproxcard_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Unpack_H10306(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
210.     memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=126
Status	New

The size of the buffer used by Pack_N10002 in hidproxmessage_t, at line 219 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_N10002 passes to hidproxmessage_t, at line 219 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	220	220
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Pack_N10002(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
220.     memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=127
Status	New

The size of the buffer used by Unpack_N10002 in hidproxcard_t, at line 230 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_N10002 passes to hidproxcard_t, at line 230 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	231	231
Object	hidproxcard_t	hidproxcard_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Unpack_N10002(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
231.    memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=128
Status	New

The size of the buffer used by Pack_C1k35s in hidproxmessage_t, at line 238 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_C1k35s passes to hidproxmessage_t, at line 238 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	239	239
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Pack_C1k35s(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
239.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=129
Status	New

The size of the buffer used by Unpack_C1k35s in hidproxcard_t, at line 253 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_C1k35s passes to hidproxcard_t, at line 253 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	254	254
Object	hidproxcard_t	hidproxcard_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Unpack_C1k35s(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
254.    memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=130
Status	New

The size of the buffer used by Pack_H10320 in hidproxmessage_t, at line 265 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_H10320 passes to hidproxmessage_t, at line 265 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	266	266
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Pack_H10320(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
266.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=131
Status	New

The size of the buffer used by Unpack_H10320 in hidproxcard_t, at line 290 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_H10320 passes to hidproxcard_t, at line 290 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	291	291
Object	hidproxcard_t	hidproxcard_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Unpack_H10320(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
291.    memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=132
Status	New

The size of the buffer used by Pack_S12906 in hidproxmessage_t, at line 312 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_S12906 passes to hidproxmessage_t, at line 312 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	313	313
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet

File Name	proxmark3/hidcardformats.c

Method bool Pack_S12906(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
313.     memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=133
Status	New

The size of the buffer used by Unpack_S12906 in hidproxcard_t, at line 330 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_S12906 passes to hidproxcard_t, at line 330 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	331	331
Object	hidproxcard_t	hidproxcard_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Unpack_S12906(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
331.     memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=134
Status	New

The size of the buffer used by Pack_Sie36 in hidproxmessage_t, at line 342 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_Sie36 passes to hidproxmessage_t, at line 342 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	343	343
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Pack_Sie36(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
343.     memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=135
Status	New

The size of the buffer used by Unpack_Sie36 in hidproxcard_t, at line 359 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_Sie36 passes to hidproxcard_t, at line 359 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	360	360
Object	hidproxcard_t	hidproxcard_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Unpack_Sie36(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
360.     memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=136
Status	New

The size of the buffer used by Pack_C15001 in hidproxmessage_t, at line 370 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_C15001 passes to hidproxmessage_t, at line 370 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	371	371
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Pack_C15001(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
371.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=137
Status	New

The size of the buffer used by Unpack_C15001 in hidproxcard_t, at line 388 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_C15001 passes to hidproxcard_t, at line 388 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	389	389
Object	hidproxcard_t	hidproxcard_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Unpack_C15001(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
389.    memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=138
Status	New

The size of the buffer used by Pack_H10302 in hidproxmessage_t, at line 400 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_H10302 passes to hidproxmessage_t, at line 400 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	401	401
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Pack_H10302(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
401.      memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=139
Status	New

The size of the buffer used by Unpack_H10302 in hidproxcard_t, at line 416 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_H10302 passes to hidproxcard_t, at line 416 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	417	417
Object	hidproxcard_t	hidproxcard_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Unpack_H10302(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
417.      memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=140
Status	New

The size of the buffer used by Pack_H10304 in hidproxmessage_t, at line 426 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_H10304 passes to hidproxmessage_t, at line 426 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	427	427
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Pack_H10304(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
427.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=141
Status	New

The size of the buffer used by Unpack_H10304 in hidproxcard_t, at line 440 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_H10304 passes to hidproxcard_t, at line 440 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	441	441
Object	hidproxcard_t	hidproxcard_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Unpack_H10304(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
441.    memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=142
Status	New

The size of the buffer used by Pack_P10001 in hidproxmessage_t, at line 451 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_P10001 passes to hidproxmessage_t, at line 451 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	452	452

Object	hidproxmessage_t	hidproxmessage_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Pack_P10001(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
452.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=143
Status	New

The size of the buffer used by Unpack_P10001 in hidproxcard_t, at line 469 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_P10001 passes to hidproxcard_t, at line 469 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	470	470
Object	hidproxcard_t	hidproxcard_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool Unpack_P10001(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
470.    memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=144
Status	New

The size of the buffer used by Pack_C1k48s in hidproxmessage_t, at line 483 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Pack_C1k48s passes to hidproxmessage_t, at line 483 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c

Line	484	484
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Pack_C1k48s(/*in*/hidproxcard_t* card, /*out*/hidproxmessage_t* packed){

```
....
484.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=145
Status	New

The size of the buffer used by Unpack_C1k48s in hidproxcard_t, at line 498 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Unpack_C1k48s passes to hidproxcard_t, at line 498 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	499	499
Object	hidproxcard_t	hidproxcard_t

Code Snippet

File Name	proxmark3/hidcardformats.c
Method	bool Unpack_C1k48s(/*in*/hidproxmessage_t* packed, /*out*/hidproxcard_t* card){

```
....
499.    memset(card, 0, sizeof(hidproxcard_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=146
Status	New

The size of the buffer used by HIDPack in hidproxmessage_t, at line 568 of proxmark3/hidcardformats.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that HIDPack passes to hidproxmessage_t, at line 568 of proxmark3/hidcardformats.c, to overwrite the target buffer.

	Source	Destination

File	proxmark3/hidcardformats.c	proxmark3/hidcardformats.c
Line	569	569
Object	hidproxmessage_t	hidproxmessage_t

Code Snippet
File Name proxmark3/hidcardformats.c
Method bool HIDPack(/* in */int FormatIndex, /* in */hidproxcard_t* card, /* out */hidproxmessage_t* packed){

```
....
569.    memset(packed, 0, sizeof(hidproxmessage_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=147
Status	New

The size of the buffer used by luaO_chunkid in l, at line 252 of proxmark3/lobject.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaO_chunkid passes to l, at line 252 of proxmark3/lobject.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/lobject.c	proxmark3/lobject.c
Line	256	256
Object	l	l

Code Snippet
File Name proxmark3/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
256.        memcpy(out, source + 1, l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=148
Status	New

The size of the buffer used by luaO_chunkid in char, at line 252 of proxmark3/lobject.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaO_chunkid passes to char, at line 252 of proxmark3/lobject.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/lobject.c	proxmark3/lobject.c

Line	256	256
Object	char	char

Code Snippet
File Name proxmark3/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
256.        memcpy(out, source + 1, l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=149
Status	New

The size of the buffer used by luaO_chunkid in l, at line 252 of proxmark3/lobject.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaO_chunkid passes to l, at line 252 of proxmark3/lobject.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/lobject.c	proxmark3/lobject.c
Line	264	264
Object	l	l

Code Snippet
File Name proxmark3/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
264.        memcpy(out, source + 1, l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=150
Status	New

The size of the buffer used by luaO_chunkid in char, at line 252 of proxmark3/lobject.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaO_chunkid passes to char, at line 252 of proxmark3/lobject.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/lobject.c	proxmark3/lobject.c
Line	264	264

Object	char	char

Code Snippet
File Name proxmark3/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
264.         memcpy(out, source + 1, l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=151
Status	New

The size of the buffer used by luaO_chunkid in bufflen, at line 252 of proxmark3/lobject.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaO_chunkid passes to bufflen, at line 252 of proxmark3/lobject.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/lobject.c	proxmark3/lobject.c
Line	268	268
Object	bufflen	bufflen

Code Snippet
File Name proxmark3/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
268.         memcpy(out, source + 1 + l - bufflen, bufflen *
sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=152
Status	New

The size of the buffer used by luaO_chunkid in char, at line 252 of proxmark3/lobject.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaO_chunkid passes to char, at line 252 of proxmark3/lobject.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/lobject.c	proxmark3/lobject.c
Line	268	268
Object	char	char

Code Snippet

File Name	proxmark3/lobject.c
Method	void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
268.        memcpy(out, source + 1 + l - bufflen, bufflen *
sizeof(char));
```

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=1050058&projectid=50048&pathid=79
Status	New

The application performs an illegal operation in CmdHF14AMfNested, in proxmark3/cmdhfmf.c. In line 597, the program attempts to divide by iterations, 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 iterations in CmdHF14AMfNested of proxmark3/cmdhfmf.c, at line 597.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	824	824
Object	iterations	iterations

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfNested(const char *Cmd) {

```
....
824.            PrintAndLog("Time in nested: %1.3f (%1.3f sec per
key)", ((float)(msclock() - msclock1))/1000.0, ((float)(msclock() -
msclock1))/iterations/1000.0);
```

Divide By Zero\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=80
Status	New

The application performs an illegal operation in pskFindFirstPhaseShift, in proxmark3/lfdemod.c. In line 281, the program attempts to divide by waveLenCnt, 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 waveLenCnt in pskFindFirstPhaseShift of proxmark3/lfdemod.c, at line 281.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	293	293
Object	waveLenCnt	waveLenCnt

Code Snippet
File Name proxmark3/lfdemod.c
Method size_t pskFindFirstPhaseShift(uint8_t samples[], size_t size, uint8_t *curPhase, size_t waveStart, uint16_t fc, uint16_t *fullWaveLen) {

```
....
293.                           lastAvgWaveVal = avgWaveVal/(waveLenCnt);
```

Divide By Zero\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=81
Status	New

The application performs an illegal operation in DetectStrongNRZClk, in proxmark3/lfdemod.c. In line 508, the program attempts to divide by size, 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 size in DetectStrongNRZClk of proxmark3/lfdemod.c, at line 508.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	538	538
Object	size	size

Code Snippet
File Name proxmark3/lfdemod.c
Method int DetectStrongNRZClk(uint8_t *dest, size_t size, int peak, int low, bool *strong) {

```
....
538.         if (transitionSampleCount / size < 10) {
```

Divide By Zero\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=82
Status	New

The application performs an illegal operation in DetectST, in proxmark3/lfdemod.c. In line 979, the program attempts to divide by clk, 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 clk in DetectST of proxmark3/lfdemod.c, at line 979.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	1031	1031
Object	clk	clk

Code Snippet
File Name proxmark3/lfdemod.c
Method bool DetectST(uint8_t buffer[], size_t *size, int *foundclock, size_t *ststart, size_t *stend) {

```
....
1031.       if (g_debugMode==2) prnt("DEBUG STT: start of data: %d end
of data: %d, datalen: %d, clk: %d, bits: %d, phaseoff: %d", skip, end,
end-skip, clk, (end-skip)/clk, phaseoff);
```

Divide By Zero\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=83
Status	New

The application performs an illegal operation in DetectST, in proxmark3/lfdemod.c. In line 979, the program attempts to divide by clk, 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 clk in DetectST of proxmark3/lfdemod.c, at line 979.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	1036	1036
Object	clk	clk

Code Snippet
File Name proxmark3/lfdemod.c
Method bool DetectST(uint8_t buffer[], size_t *size, int *foundclock, size_t *ststart, size_t *stend) {

```
....
1036.       if ( clk - (datalen % clk) <= clk/8) {
```

Divide By Zero\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=500

	48&pathid=84
Status	New

The application performs an illegal operation in DetectST, in proxmark3/lfdemod.c. In line 979, the program attempts to divide by clk, 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 clk in DetectST of proxmark3/lfdemod.c, at line 979.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	1038	1038
Object	clk	clk

Code Snippet
File Name proxmark3/lfdemod.c
Method bool DetectST(uint8_t buffer[], size_t *size, int *foundclock, size_t *ststart, size_t *stend) {

```
....
1038.               datalen += clk - (datalen % clk);
```

Divide By Zero\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=85
Status	New

The application performs an illegal operation in DetectST, in proxmark3/lfdemod.c. In line 979, the program attempts to divide by clk, 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 clk in DetectST of proxmark3/lfdemod.c, at line 979.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	1039	1039
Object	clk	clk

Code Snippet
File Name proxmark3/lfdemod.c
Method bool DetectST(uint8_t buffer[], size_t *size, int *foundclock, size_t *ststart, size_t *stend) {

```
....
1039.        } else if ( (datalen % clk) <= clk/8 ) {
```

Divide By Zero\Path 8:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=86
Status	New

The application performs an illegal operation in DetectST, in proxmark3/lfdemod.c. In line 979, the program attempts to divide by clk, 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 clk in DetectST of proxmark3/lfdemod.c, at line 979.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	1041	1041
Object	clk	clk

Code Snippet

File Name proxmark3/lfdemod.c

Method bool DetectST(uint8_t buffer[], size_t *size, int *foundclock, size_t *ststart, size_t *stend) {

```
....
1041.            datalen -= datalen % clk;
```

Divide By Zero\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=87
Status	New

The application performs an illegal operation in DetectST, in proxmark3/lfdemod.c. In line 979, the program attempts to divide by clk, 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 clk in DetectST of proxmark3/lfdemod.c, at line 979.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	1043	1043
Object	clk	clk

Code Snippet

File Name proxmark3/lfdemod.c

Method bool DetectST(uint8_t buffer[], size_t *size, int *foundclock, size_t *ststart, size_t *stend) {

```
....
1043.              if (g_debugMode==2) prnt("DEBUG STT: datalen not
divisible by clk: %u %% %d = %d - quitting", datalen, clk, datalen %
clk);
```

Divide By Zero\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=88
Status	New

The application performs an illegal operation in DetectST, in proxmark3/lfdemod.c. In line 979, the program attempts to divide by clk, 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 clk in DetectST of proxmark3/lfdemod.c, at line 979.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	1047	1047
Object	clk	clk

Code Snippet	
File Name	proxmark3/lfdemod.c
Method	bool DetectST(uint8_t buffer[], size_t *size, int *foundclock, size_t *ststart, size_t *stend) {

```
....
1047.        if (datalen/clk < 8*4) {
```

Heap Inspection

Query Path:
CPP\Cx\CPP Medium Threat\Heap Inspection Version:1

Categories

OWASP Top 10 2013: A6-Sensitive Data Exposure
FISMA 2014: Media Protection
NIST SP 800-53: SC-4 Information in Shared Resources (P1)
OWASP Top 10 2017: A3-Sensitive Data Exposure

Description

Heap Inspection\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=455
Status	New

Method CmdT55xxReadBlock at line 351 of proxmark3/cmdlft55xx.c defines password, which is designated to contain user passwords. However, while plaintext passwords are later assigned to password, this variable is never cleared from memory.

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	353	353
Object	password	password

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method int CmdT55xxReadBlock(const char *Cmd) {

```
....
353.        uint32_t password = 0; //default to blank Block 7
```

Heap Inspection\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=456
Status	New

Method CmdT55xxDetect at line 498 of proxmark3/cmdlft55xx.c defines password, which is designated to contain user passwords. However, while plaintext passwords are later assigned to password, this variable is never cleared from memory.

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	502	502
Object	password	password

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method int CmdT55xxDetect(const char *Cmd){

```
....
502.        uint32_t password = 0;
```

Heap Inspection\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=457
Status	New

Method CmdT55xxWakeUp at line 929 of proxmark3/cmdlft55xx.c defines password, which is designated to contain user passwords. However, while plaintext passwords are later assigned to password, this variable is never cleared from memory.

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	930	930
Object	password	password

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method int CmdT55xxWakeUp(const char *Cmd) {

```
....
930.        uint32_t password = 0;
```

Heap Inspection\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=458
Status	New

Method CmdT55xxWriteBlock at line 944 of proxmark3/cmdlft55xx.c defines password, which is designated to contain user passwords. However, while plaintext passwords are later assigned to password, this variable is never cleared from memory.

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	947	947
Object	password	password

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method int CmdT55xxWriteBlock(const char *Cmd) {

```
....
947.        uint32_t password = 0; //default to blank Block 7
```

Heap Inspection\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=459
Status	New

Method CmdT55xxDump at line 1256 of proxmark3/cmdlft55xx.c defines password, which is designated to contain user passwords. However, while plaintext passwords are later assigned to password, this variable is never cleared from memory.

	Source	Destination

File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1258	1258
Object	password	password

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method int CmdT55xxDump(const char *Cmd){

```
....
1258.        uint32_t password = 0;
```

Heap Inspection\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=460
Status	New

Method CmdT55xxBruteForce at line 1452 of proxmark3/cmdlft55xx.c defines start_password, which is designated to contain user passwords. However, while plaintext passwords are later assigned to start_password, this variable is never cleared from memory.

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1461	1461
Object	start_password	start_password

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method int CmdT55xxBruteForce(const char *Cmd) {

```
....
1461.        uint32_t start_password = 0x00000000; //start password
```

Heap Inspection\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=461
Status	New

Method CmdT55xxBruteForce at line 1452 of proxmark3/cmdlft55xx.c defines end_password, which is designated to contain user passwords. However, while plaintext passwords are later assigned to end_password, this variable is never cleared from memory.

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c

Line	1462	1462
Object	end_password	end_password

Code Snippet
File Name	proxmark3/cmdlft55xx.c
Method	int CmdT55xxBruteForce(const char *Cmd) {

```
....
1462.        uint32_t end_password   = 0xFFFFFFFF; //end   password
```

Heap Inspection\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=462
Status	New

Method CmdT55xxDetectPage1 at line 1746 of proxmark3/cmdlft55xx.c defines password, which is designated to contain user passwords. However, while plaintext passwords are later assigned to password, this variable is never cleared from memory.

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1750	1750
Object	password	password

Code Snippet
File Name	proxmark3/cmdlft55xx.c
Method	int CmdT55xxDetectPage1(const char *Cmd){

```
....
1750.        uint32_t password = 0;
```

Heap Inspection\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=463
Status	New

Method CmdT55xxBruteForce at line 1452 of proxmark3/cmdlft55xx.c defines testpwd, which is designated to contain user passwords. However, while plaintext passwords are later assigned to testpwd, this variable is never cleared from memory.

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1540	1540

Object	testpwd	testpwd

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method int CmdT55xxBruteForce(const char *Cmd) {

```
....
1540.           uint64_t testpwd = 0x00;
```

Heap Inspection\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=464
Status	New

Method CmdT55xxInfo at line 1190 of proxmark3/cmdlft55xx.c defines pwd, which is designated to contain user passwords. However, while plaintext passwords are later assigned to pwd, this variable is never cleared from memory.

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1224	1224
Object	pwd	pwd

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method int CmdT55xxInfo(const char *Cmd){

```
....
1224.       uint32_t pwd      = PackBits(si, 1, DemodBuffer); si += 1;
```

Stored Buffer Overflow boundcpy

Query Path:
CPP\Cx\CPP Stored Vulnerabilities\Stored Buffer Overflow boundcpy Version:1

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)
OWASP Top 10 2017: A1-Injection

Description

Stored Buffer Overflow boundcpy\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=475
Status	New

The size of the buffer used by CmdHF14AMfSim in buf, at line 1489 of proxmark3/cmdhfmf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that CmdHF14AMfSim passes to buf, at line 1489 of proxmark3/cmdhfmf.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1601	1598
Object	buf	buf

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfSim(const char *Cmd) {

```
....
1601.                   if (fgets(buf, sizeof(buf), f) == NULL) {
....
1598.                   memset(buf, 0, sizeof(buf));
```

Stored Buffer Overflow boundcpy\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=476
Status	New

The size of the buffer used by CmdHF14AMfSim in sizeof, at line 1489 of proxmark3/cmdhfmf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that CmdHF14AMfSim passes to buf, at line 1489 of proxmark3/cmdhfmf.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1601	1598
Object	buf	sizeof

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfSim(const char *Cmd) {

```
....
1601.                   if (fgets(buf, sizeof(buf), f) == NULL) {
....
1598.                   memset(buf, 0, sizeof(buf));
```

Stored Buffer Overflow boundcpy\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=477

Status	New

The size of the buffer used by CmdHF14AMfELoad in buf, at line 1779 of proxmark3/cmdhfmf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that CmdHF14AMfELoad passes to buf, at line 1779 of proxmark3/cmdhfmf.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1832	1830
Object	buf	buf

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfELoad(const char *Cmd)

```
....
1832.            if (fgets(buf, sizeof(buf), f) == NULL) {
....
1830.            memset(buf, 0, sizeof(buf));
```

Stored Buffer Overflow boundcpy\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=478
Status	New

The size of the buffer used by CmdHF14AMfELoad in sizeof, at line 1779 of proxmark3/cmdhfmf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that CmdHF14AMfELoad passes to buf, at line 1779 of proxmark3/cmdhfmf.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1832	1830
Object	buf	sizeof

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfELoad(const char *Cmd)

```
....
1832.            if (fgets(buf, sizeof(buf), f) == NULL) {
....
1830.            memset(buf, 0, sizeof(buf));
```

Stored Buffer Overflow boundcpy\Path 5:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=479
Status	New

The size of the buffer used by CmdHF14AMfCLoad in buf, at line 2258 of proxmark3/cmdhfmf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that CmdHF14AMfCLoad passes to buf, at line 2258 of proxmark3/cmdhfmf.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2330	2328
Object	buf	buf

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfCLoad(const char *Cmd)

```
....
2330.                     if (fgets(buf, sizeof(buf), f) == NULL) {
....
2328.                     memset(buf, 0, sizeof(buf));
```

Stored Buffer Overflow boundcpy\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=480
Status	New

The size of the buffer used by CmdHF14AMfCLoad in sizeof, at line 2258 of proxmark3/cmdhfmf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that CmdHF14AMfCLoad passes to buf, at line 2258 of proxmark3/cmdhfmf.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2330	2328
Object	buf	sizeof

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfCLoad(const char *Cmd)

```
....
2330.                     if (fgets(buf, sizeof(buf), f) == NULL) {
....
2328.                     memset(buf, 0, sizeof(buf));
```

Stored Buffer Overflow boundcpy\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=481
Status	New

The size of the buffer used by loadTraceCard in buf, at line 795 of proxmark3/mifarehost.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that loadTraceCard passes to buf, at line 795 of proxmark3/mifarehost.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	817	816
Object	buf	buf

Code Snippet
File Name proxmark3/mifarehost.c
Method int loadTraceCard(uint8_t *tuid) {

```
....
817.                if (fgets(buf, sizeof(buf), f) == NULL) {
....
816.                memset(buf, 0, sizeof(buf));
```

Stored Buffer Overflow boundcpy\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=482
Status	New

The size of the buffer used by loadTraceCard in sizeof, at line 795 of proxmark3/mifarehost.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that loadTraceCard passes to buf, at line 795 of proxmark3/mifarehost.c, to overwrite the target buffer.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	817	816
Object	buf	sizeof

Code Snippet
File Name proxmark3/mifarehost.c
Method int loadTraceCard(uint8_t *tuid) {

```
....
817.                if (fgets(buf, sizeof(buf), f) == NULL) {
....
816.                memset(buf, 0, sizeof(buf));
```

Buffer Overflow AddressOfLocalVarReturned

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

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.2 - Buffer overflows
NIST SP 800-53: SC-5 Denial of Service Protection (P1)
OWASP Top 10 2017: A1-Injection

Description

Buffer Overflow AddressOfLocalVarReturned\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=96
Status	New

The pointer candidates at proxmark3/crapto1.c in line 441 is being used after it has been freed.

	Source	Destination
File	proxmark3/crapto1.c	proxmark3/crapto1.c
Line	461	461
Object	candidates	candidates

Code Snippet
File Name proxmark3/crapto1.c
Method uint32_t *lfsr_prefix_ks(uint8_t ks[8], int isodd)

```
....
461.        return candidates;
```

Buffer Overflow AddressOfLocalVarReturned\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=97
Status	New

The pointer fndClk at proxmark3/lfdemod.c in line 223 is being used after it has been freed.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	228	228
Object	fndClk	fndClk

Code Snippet
File Name proxmark3/lfdemod.c

Method	int getClosestClock(int testclk) {

```
....
228.                    return fndClk[clkCnt];
```

Buffer Overflow AddressOfLocalVarReturned\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=98
Status	New

The pointer bestStart at proxmark3/lfdemod.c in line 412 is being used after it has been freed.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	505	505
Object	bestStart	bestStart

Code Snippet

File Name	proxmark3/lfdemod.c
Method	int DetectASKClock(uint8_t dest[], size_t size, int *clock, int maxErr) {

```
....
505.          return bestStart[best];
```

Buffer Overflow AddressOfLocalVarReturned\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=99
Status	New

The pointer clk at proxmark3/lfdemod.c in line 547 is being used after it has been freed.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	660	660
Object	clk	clk

Code Snippet

File Name	proxmark3/lfdemod.c
Method	int DetectNRZClock(uint8_t dest[], size_t size, int clock, size_t *clockStartIdx) {

```
....
660.          return clk[best];
```

Buffer Overflow AddressOfLocalVarReturned\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=100
Status	New

The pointer clk at proxmark3/lfdemod.c in line 848 is being used after it has been freed.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	951	951
Object	clk	clk

Code Snippet

File Name	proxmark3/lfdemod.c
Method	uint8_t detectFSKClk(uint8_t *BitStream, size_t size, uint8_t fcHigh, uint8_t fcLow, int *firstClockEdge) {

```
....
951.          return clk[ii];
```

Buffer Overflow AddressOfLocalVarReturned\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=101
Status	New

The pointer fcLens at proxmark3/lfdemod.c in line 667 is being used after it has been freed.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	749	749
Object	fcLens	fcLens

Code Snippet

File Name	proxmark3/lfdemod.c
Method	uint16_t countFC(uint8_t *BitStream, size_t size, uint8_t fskAdj) {

```
....
749.          return (uint16_t)fcLens[best2] << 8 | fcLens[best1];
```

Buffer Overflow AddressOfLocalVarReturned\Path 7:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=102
Status	New

The pointer fcLens at proxmark3/lfdemod.c in line 667 is being used after it has been freed.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	749	749
Object	fcLens	fcLens

Code Snippet
File Name proxmark3/lfdemod.c
Method uint16_t countFC(uint8_t *BitStream, size_t size, uint8_t fskAdj) {

```
....
749.        return (uint16_t)fcLens[best2] << 8 | fcLens[best1];
```

MemoryFree on StackVariable

Query Path:
CPP\Cx\CPP Medium Threat\MemoryFree on StackVariable Version:0

Description

MemoryFree on StackVariable\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=166
Status	New

Calling free() (line 505) on a variable that was not dynamically allocated (line 505) in file proxmark3/crapto1.c may result with a crash.

	Source	Destination
File	proxmark3/crapto1.c	proxmark3/crapto1.c
Line	530	530
Object	odd	odd

Code Snippet
File Name proxmark3/crapto1.c
Method lfsr_common_prefix(uint32_t pfx, uint32_t rr, uint8_t ks[8], uint8_t par[8][8], uint32_t no_par)

```
....
530.        free(odd);
```

MemoryFree on StackVariable\Path 2:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=167
Status	New

Calling free() (line 505) on a variable that was not dynamically allocated (line 505) in file proxmark3/crapto1.c may result with a crash.

	Source	Destination
File	proxmark3/crapto1.c	proxmark3/crapto1.c
Line	531	531
Object	even	even

Code Snippet

File Name	proxmark3/crapto1.c
Method	lfsr_common_prefix(uint32_t pfx, uint32_t rr, uint8_t ks[8], uint8_t par[8][8], uint32_t no_par)

```
....
531.        free(even);
```

MemoryFree on StackVariable\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=168
Status	New

Calling free() (line 119) on a variable that was not dynamically allocated (line 119) in file proxmark3/mifarehost.c may result with a crash.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	187	187
Object	last_keylist	last_keylist

Code Snippet

File Name	proxmark3/mifarehost.c
Method	int mfDarkside(uint64_t *key) {

```
....
187.                        free(last_keylist);
```

MemoryFree on StackVariable\Path 4:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=169
Status	New

Calling free() (line 119) on a variable that was not dynamically allocated (line 119) in file proxmark3/mifarehost.c may result with a crash.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	210	210
Object	last_keylist	last_keylist

Code Snippet

File Name proxmark3/mifarehost.c

Method int mfDarkside(uint64_t *key) {

```
....
210.                     free(last_keylist);
```

MemoryFree on StackVariable\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=170
Status	New

Calling free() (line 119) on a variable that was not dynamically allocated (line 119) in file proxmark3/mifarehost.c may result with a crash.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	211	211
Object	keylist	keylist

Code Snippet

File Name proxmark3/mifarehost.c

Method int mfDarkside(uint64_t *key) {

```
....
211.                     free(keylist);
```

MemoryFree on StackVariable\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=171
Status	New

Calling free() (line 119) on a variable that was not dynamically allocated (line 119) in file proxmark3/mifarehost.c may result with a crash.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	215	215
Object	last_keylist	last_keylist

Code Snippet
File Name	proxmark3/mifarehost.c
Method	int mfDarkside(uint64_t *key) {

```
....
215.                            free(last_keylist);
```

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=1050058&projectid=50048&pathid=258
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 244 of proxmark3/lfdemod.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	271	271
Object	AssignExpr	AssignExpr

Code Snippet
File Name	proxmark3/lfdemod.c
Method	bool loadWaveCounters(uint8_t samples[], size_t size, int lowToLowWaveLen[], int highToLowWaveLen[], int *waveCnt, int *skip, int *minClk, int *high, int *low) {

```
....
271.                highToLowWaveLen[*waveCnt] = i - firstHigh; //first
high to first low
```

Integer Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=259
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 244 of proxmark3/lfdemod.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	272	272
Object	AssignExpr	AssignExpr

Code Snippet
File Name proxmark3/lfdemod.c
Method bool loadWaveCounters(uint8_t samples[], size_t size, int lowToLowWaveLen[], int highToLowWaveLen[], int *waveCnt, int *skip, int *minClk, int *high, int *low) {

```
....
272.                lowToLowWaveLen[*waveCnt] = i - firstLow;
```

Integer Overflow\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=260
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 377 of proxmark3/lfdemod.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	396	396
Object	AssignExpr	AssignExpr

Code Snippet
File Name proxmark3/lfdemod.c

Method int DetectStrongAskClock(uint8_t dest[], size_t size, int high, int low, int *clock)
{

```
....
396.                        shortestWaveIdx = startwave;
```

Integer Overflow\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=261
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 508 of proxmark3/lfdemod.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	529	529
Object	AssignExpr	AssignExpr

Code Snippet
File Name proxmark3/lfdemod.c
Method int DetectStrongNRZClk(uint8_t *dest, size_t size, int peak, int low, bool *strong)
{

```
....
529.                        if (i-transition1 < lowestTransition)
lowestTransition = i-transition1;
```

Integer Overflow\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=262
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 547 of proxmark3/lfdemod.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	606	606
Object	AssignExpr	AssignExpr

Code Snippet
File Name proxmark3/lfdemod.c

Method	int DetectNRZClock(uint8_t dest[], size_t size, int clock, size_t *clockStartIdx) {

```
....
606.                            lastBit = ii-clk[clkCnt];
```

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=1050058&projectid=50048&pathid=465
Status	New

	Source	Destination
File	proxmark3/crapto1.c	proxmark3/crapto1.c
Line	230	230
Object	odd_tail	odd_tail

Code Snippet	
File Name	proxmark3/crapto1.c
Method	struct Crypto1State* lfsr_recovery32(uint32_t ks2, uint32_t in)

```
....
230.          odd_head = odd_tail = malloc(sizeof(uint32_t) << 21);
```

Memory Leak\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=466
Status	New

	Source	Destination
File	proxmark3/crapto1.c	proxmark3/crapto1.c
Line	231	231
Object	even_tail	even_tail

Code Snippet	
File Name	proxmark3/crapto1.c
Method	struct Crypto1State* lfsr_recovery32(uint32_t ks2, uint32_t in)

```
....
231.         even_head = even_tail = malloc(sizeof(uint32_t) << 21);
```

Memory Leak\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=467
Status	New

	Source	Destination
File	proxmark3/crapto1.c	proxmark3/crapto1.c
Line	308	308
Object	statelist	statelist

Code Snippet
File Name proxmark3/crapto1.c
Method struct Crypto1State* lfsr_recovery64(uint32_t ks2, uint32_t ks3)

```
....
308.         sl = statelist = malloc(sizeof(struct Crypto1State) << 4);
```

Memory Leak\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=468
Status	New

	Source	Destination
File	proxmark3/crapto1.c	proxmark3/crapto1.c
Line	418	418
Object	dist	dist

Code Snippet
File Name proxmark3/crapto1.c
Method int nonce_distance(uint32_t from, uint32_t to)

```
....
418.               dist = malloc(2 << 16);
```

Memory Leak\Path 5:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=469
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	445	445
Object	states	states

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method float brute_force_benchmark()

```
....
445.        test_candidates[0].states[ODD_STATE] =
malloc((TEST_BENCH_SIZE+1) * sizeof(uint32_t));
```

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=1050058&projectid=50048&pathid=471
Status	New

The variable declared in keylist at proxmark3/mifarehost.c in line 119 is not initialized when it is used by keylist at proxmark3/mifarehost.c in line 119.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	123	184
Object	keylist	keylist

Code Snippet
File Name proxmark3/mifarehost.c
Method int mfDarkside(uint64_t *key) {

```
....
123.        uint64_t *keylist = NULL, *last_keylist = NULL;
....
184.                    qsort(keylist, keycount, sizeof(*keylist),
compare_uint64);
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=472
Status	New

The variable declared in Pointer at proxmark3/mifarehost.c in line 80 is not initialized when it is used by keylist at proxmark3/mifarehost.c in line 119.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	101	184
Object	Pointer	keylist

Code Snippet

File Name proxmark3/mifarehost.c

Method static uint32_t nonce2key(uint32_t uid, uint32_t nt, uint32_t nr, uint32_t ar, uint64_t par_info, uint64_t ks_info, uint64_t **keys) {

```
....
101.            *keys = NULL;
```

File Name proxmark3/mifarehost.c

Method int mfDarkside(uint64_t *key) {

```
....
184.                    qsort(keylist, keycount, sizeof(*keylist),
compare_uint64);
```

Use of Zero Initialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=473
Status	New

The variable declared in keylist at proxmark3/mifarehost.c in line 119 is not initialized when it is used by keylist at proxmark3/mifarehost.c in line 119.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	123	201
Object	keylist	keylist

Code Snippet

File Name	proxmark3/mifarehost.c
Method	int mfDarkside(uint64_t *key) {

```
....
123.          uint64_t *keylist = NULL, *last_keylist = NULL;
....
201.                        num_to_bytes(keylist[i], 6, keys_to_chk+i);
```

Use of Zero Initialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=474
Status	New

The variable declared in Pointer at proxmark3/mifarehost.c in line 80 is not initialized when it is used by keylist at proxmark3/mifarehost.c in line 119.

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	101	201
Object	Pointer	keylist

Code Snippet

File Name	proxmark3/mifarehost.c
Method	static uint32_t nonce2key(uint32_t uid, uint32_t nt, uint32_t nr, uint32_t ar, uint64_t par_info, uint64_t ks_info, uint64_t **keys) {

```
....
101.                 *keys = NULL;
```

▼

File Name	proxmark3/mifarehost.c
Method	int mfDarkside(uint64_t *key) {

```
....
201.                        num_to_bytes(keylist[i], 6, keys_to_chk+i);
```

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=1050058&projectid=50048&pathid=172
Status	New

The function itemsize in proxmark3/elite_crack.c at line 500 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	proxmark3/elite_crack.c	proxmark3/elite_crack.c
Line	507	507
Object	itemsize	itemsize

Code Snippet

File Name	proxmark3/elite_crack.c
Method	int bruteforceDump(uint8_t dump[], size_t dumpsize, uint16_t keytable[])

```
....
507.        dumpdata* attack = (dumpdata* ) malloc(itemsize);
```

Wrong Size t Allocation\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=173
Status	New

The function fsize in proxmark3/elite_crack.c at line 542 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	proxmark3/elite_crack.c	proxmark3/elite_crack.c
Line	561	561
Object	fsize	fsize

Code Snippet

File Name	proxmark3/elite_crack.c
Method	int bruteforceFile(const char *filename, uint16_t keytable[])

```
....
561.        uint8_t *dump = malloc(fsize);
```

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=1050058&projectid=50048&pathid=453
Status	New

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	210	187
Object	last_keylist	last_keylist

Code Snippet
File Name proxmark3/mifarehost.c
Method int mfDarkside(uint64_t *key) {

```
....
210.                        free(last_keylist);
....
187.                              free(last_keylist);
```

Double Free\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=454
Status	New

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	210	215
Object	last_keylist	last_keylist

Code Snippet
File Name proxmark3/mifarehost.c
Method int mfDarkside(uint64_t *key) {

```
....
210.                        free(last_keylist);
....
215.                        free(last_keylist);
```

Use of Uninitialized Variable

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

Categories

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

Description

Use of Uninitialized Variable\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=470
Status	New

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	1551	1612
Object	avgWaveVal	avgWaveVal

Code Snippet
File Name proxmark3/lfdemod.c
Method int pskRawDemod_ext(uint8_t dest[], size_t *size, int *clock, int *invert, int *startIdx) {

```
....
1551.       uint16_t fullWaveLen=0, waveLenCnt=0, avgWaveVal;
....
1612.                               avgWaveVal += dest[i+1];
```

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=1050058&projectid=50048&pathid=174
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 1288.

	Source	Destination

File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1302
Object	null	Y

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_select_comb(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1302.           MBEDTLS_MPI_CHK( mbedtls_mpi_safe_cond_assign( &R->Y,
&T[j].Y, j == ii ) );
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=175
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by X at proxmark3/ecp.c in line 372.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	376
Object	null	X

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method int mbedtls_ecp_copy(mbedtls_ecp_point *P, const mbedtls_ecp_point *Q)

```
....
376.        MBEDTLS_MPI_CHK( mbedtls_mpi_copy( &P->X, &Q->X ) );
```

NULL Pointer Dereference\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=176
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by P at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1064
Object	null	P

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1064.       MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &T1,  &T1,    &P->X )
);  MOD_SUB( T1 );
```

NULL Pointer Dereference\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=177
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by P at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1065

Object	null	P

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name proxmark3/ecp.c

Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1065.       MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &T2,  &T2,    &P->Y )
);  MOD_SUB( T2 );
```

NULL Pointer Dereference\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=178
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by P at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1093
Object	null	P

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name proxmark3/ecp.c

Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1093.       MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &Y,   &T3,    &T4    )
);  MOD_SUB( Y  );
```

NULL Pointer Dereference\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=179
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by grp at proxmark3/ecp.c in line 672.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	686
Object	null	grp

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_modp(mbedtls_mpi *N, const mbedtls_ecp_group *grp)

```
....
686.      MBEDTLS_MPI_CHK( grp->modp( N ) );
```

NULL Pointer Dereference\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=180
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by P at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	989
Object	null	P

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
989.       MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &S,   &S,     &T      )
); MOD_SUB( S );
```

NULL Pointer Dereference\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=181
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by P at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	991
Object	null	P

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
991.       MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &S,   &S,     &U      )
); MOD_SUB( S );
```

NULL Pointer Dereference\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=182

Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	998
Object	null	Y

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c
Method static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
998.      MBEDTLS_MPI_CHK( mbedtls_mpi_copy( &R->Y, &S ) );
```

NULL Pointer Dereference\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=183
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Z at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	999
Object	null	Z

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name	proxmark3/ecp.c
Method	static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
999.        MBEDTLS_MPI_CHK( mbedtls_mpi_copy( &R->Z, &U ) );
```

NULL Pointer Dereference\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=184
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Z at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	994
Object	null	Z

Code Snippet

File Name	proxmark3/ecp.c
Method	static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.        T = p_eq_g ? grp->T : NULL;
```

File Name	proxmark3/ecp.c
Method	static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
994.        MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &U,   &P->Y,   &P->Z    )
); MOD_MUL( U );
```

NULL Pointer Dereference\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=185
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 927.

	Source	Destination

File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	994
Object	null	Y

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
994.       MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &U,   &P->Y,   &P->Z    )
); MOD_MUL( U );
```

NULL Pointer Dereference\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=186
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by X at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	997
Object	null	X

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
997.       MBEDTLS_MPI_CHK( mbedtls_mpi_copy( &R->X, &T ) );
```

NULL Pointer Dereference\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=187
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by P at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	986
Object	null	P

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.      T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
986.       MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &T,   &T,      &S       )
); MOD_SUB( T );
```

NULL Pointer Dereference\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=188
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by P at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	985

Object	null	P

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name proxmark3/ecp.c

Method static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
985.        MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &T,   &T,      &S       )
); MOD_SUB( T );
```

NULL Pointer Dereference\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=189
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by X at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	976
Object	null	X

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name proxmark3/ecp.c

Method static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
976.        MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &S,   &P->X,   &T       )
); MOD_MUL( S );
```

NULL Pointer Dereference\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=190
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Z at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1082
Object	null	Z

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1082.       MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &Z,    &P->Z,   &T1    )
);   MOD_MUL( Z   );
```

NULL Pointer Dereference\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=191
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Z at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1061
Object	null	Z

Code Snippet
File Name proxmark3/ecp.c

Method	static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name	proxmark3/ecp.c
Method	static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1061.       MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &T2,  &T1,    &P->Z )
);  MOD_MUL( T2 );
```

NULL Pointer Dereference\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=192
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Z at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1060
Object	null	Z

Code Snippet

File Name	proxmark3/ecp.c
Method	static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name	proxmark3/ecp.c
Method	static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1060.       MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &T1,  &P->Z,  &P->Z )
);  MOD_MUL( T1 );
```

NULL Pointer Dereference\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=500

	48&pathid=193
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Z at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1060
Object	null	Z

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1060.       MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &T1,  &P->Z,  &P->Z )
);  MOD_MUL( T1 );
```

NULL Pointer Dereference\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=194
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Z at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1045
Object	null	Z

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name	proxmark3/ecp.c
Method	static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1045.       if( mbedtls_mpi_cmp_int( &P->Z, 0 ) == 0 )
```

NULL Pointer Dereference\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=195
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Z at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1097
Object	null	Z

Code Snippet

File Name	proxmark3/ecp.c
Method	static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name	proxmark3/ecp.c
Method	static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1097.       MBEDTLS_MPI_CHK( mbedtls_mpi_copy( &R->Z, &Z ) );
```

NULL Pointer Dereference\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=196
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by X at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1085
Object	null	X

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1085.       MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &T3,  &T3,    &P->X )
);  MOD_MUL( T3 );
```

NULL Pointer Dereference\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=197
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by X at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1064
Object	null	X

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1064.        MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &T1,   &T1,     &P->X )
);  MOD_SUB( T1 );
```

NULL Pointer Dereference\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=198
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by X at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1095
Object	null	X

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.        T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1095.        MBEDTLS_MPI_CHK( mbedtls_mpi_copy( &R->X, &X ) );
```

NULL Pointer Dereference\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=199
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by P at proxmark3/ecp.c in line 672.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	690

Object	null	P

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_modp(mbedtls_mpi *N, const mbedtls_ecp_group *grp)

```
....
690.          MBEDTLS_MPI_CHK( mbedtls_mpi_add_mpi( N, N, &grp->P ) );
```

NULL Pointer Dereference\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=200
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Z at proxmark3/ecp.c in line 372.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	378
Object	null	Z

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method int mbedtls_ecp_copy(mbedtls_ecp_point *P, const mbedtls_ecp_point *Q)

```
....
378.      MBEDTLS_MPI_CHK( mbedtls_mpi_copy( &P->Z, &Q->Z ) );
```

NULL Pointer Dereference\Path 28:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=201
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1092
Object	null	Y

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1092.       MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &T4,   &T4,     &P->Y )
);   MOD_MUL( T4 );
```

NULL Pointer Dereference\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=202
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by P at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1088
Object	null	P

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1088.       MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &X,    &X,      &T1    )
);  MOD_SUB( X  );
```

NULL Pointer Dereference\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=203
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by P at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1089
Object	null	P

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1089.       MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &X,    &X,      &T4    )
);  MOD_SUB( X  );
```

NULL Pointer Dereference\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=204

Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by P at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1090
Object	null	P

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c
Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1090.       MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &T3,  &T3,    &X    )
);  MOD_SUB( T3 );
```

NULL Pointer Dereference\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=205
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1065
Object	null	Y

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name proxmark3/ecp.c

Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1065.       MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &T2,  &T2,    &P->Y )
);  MOD_SUB( T2 );
```

NULL Pointer Dereference\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=206
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 372.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	377
Object	null	Y

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name proxmark3/ecp.c

Method int mbedtls_ecp_copy(mbedtls_ecp_point *P, const mbedtls_ecp_point *Q)

```
....
377.       MBEDTLS_MPI_CHK( mbedtls_mpi_copy( &P->Y, &Q->Y ) );
```

NULL Pointer Dereference\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=207
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 1025.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	1096
Object	null	Y

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_add_mixed(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1096.       MBEDTLS_MPI_CHK( mbedtls_mpi_copy( &R->Y, &Y ) );
```

NULL Pointer Dereference\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=208
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	974
Object	null	Y

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
974.        MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &T,   &P->Y,   &P->Y    )
); MOD_MUL( T );
```

NULL Pointer Dereference\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=209
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	974
Object	null	Y

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
974.        MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &T,   &P->Y,   &P->Y    )
); MOD_MUL( T );
```

NULL Pointer Dereference\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=210
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by P at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c

Line	1412	952
Object	null	P

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name proxmark3/ecp.c

Method static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
952.            MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &U,   &P->X,   &S
) ); MOD_SUB( U );
```

NULL Pointer Dereference\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=211
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by X at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	952
Object	null	X

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name proxmark3/ecp.c

Method static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
952.            MBEDTLS_MPI_CHK( mbedtls_mpi_sub_mpi( &U,   &P->X,   &S
) ); MOD_SUB( U );
```

NULL Pointer Dereference\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=212
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by X at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	951
Object	null	X

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
951.            MBEDTLS_MPI_CHK( mbedtls_mpi_add_mpi( &T,  &P->X,  &S
) ); MOD_ADD( T );
```

NULL Pointer Dereference\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=213
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Z at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	966
Object	null	Z

Code Snippet

File Name	proxmark3/ecp.c
Method	static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name	proxmark3/ecp.c
Method	static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
966.                MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &S,  &P->Z,  &P-
>Z    ) ); MOD_MUL( S );
```

NULL Pointer Dereference\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=214
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Z at proxmark3/ecp.c in line 927.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	966
Object	null	Z

Code Snippet

File Name	proxmark3/ecp.c
Method	static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

▼

File Name	proxmark3/ecp.c
Method	static int ecp_double_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
966.                MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &S,  &P->Z,  &P-
>Z    ) ); MOD_MUL( S );
```

NULL Pointer Dereference\Path 42:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=215
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 752.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	779
Object	null	Y

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.        T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_normalize_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *pt)

```
....
779.        MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &pt->Y,    &pt->Y,
&Zi      ) ); MOD_MUL( pt->Y );
```

NULL Pointer Dereference\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=216
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 752.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	779
Object	null	Y

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name	proxmark3/ecp.c
Method	static int ecp_normalize_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *pt)

```
....
779.       MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &pt->Y,    &pt->Y,
&Zi      ) ); MOD_MUL( pt->Y );
```

NULL Pointer Dereference\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=217
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 752.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	779
Object	null	Y

Code Snippet

File Name	proxmark3/ecp.c
Method	static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name	proxmark3/ecp.c
Method	static int ecp_normalize_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *pt)

```
....
779.       MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &pt->Y,    &pt->Y,
&Zi      ) ); MOD_MUL( pt->Y );
```

NULL Pointer Dereference\Path 45:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=218
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 752.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	778
Object	null	Y

Code Snippet
File Name	proxmark3/ecp.c
Method	static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name	proxmark3/ecp.c

Method	static int ecp_normalize_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *pt)

```
....
778.       MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &pt->Y,    &pt->Y,
&ZZi     ) ); MOD_MUL( pt->Y );
```

NULL Pointer Dereference\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=219
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 752.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	778
Object	null	Y

Code Snippet
File Name	proxmark3/ecp.c
Method	static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name	proxmark3/ecp.c
Method	static int ecp_normalize_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *pt)

```
....
778.        MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &pt->Y,    &pt->Y,
&ZZi     ) ); MOD_MUL( pt->Y );
```

NULL Pointer Dereference\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=220
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by Y at proxmark3/ecp.c in line 752.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	778
Object	null	Y

Code Snippet

File Name	proxmark3/ecp.c
Method	static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name	proxmark3/ecp.c
Method	static int ecp_normalize_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *pt)

```
....
778.        MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &pt->Y,    &pt->Y,
&ZZi     ) ); MOD_MUL( pt->Y );
```

NULL Pointer Dereference\Path 48:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=221
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by X at proxmark3/ecp.c in line 752.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	773
Object	null	X

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_normalize_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *pt)

```
....
773.        MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &pt->X,    &pt->X,
&ZZi     ) ); MOD_MUL( pt->X );
```

NULL Pointer Dereference\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=222
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by X at proxmark3/ecp.c in line 752.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	773
Object	null	X

Code Snippet
File Name proxmark3/ecp.c
Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_normalize_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *pt)

```
....
773.       MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &pt->X,    &pt->X,
&ZZi    ) ); MOD_MUL( pt->X );
```

NULL Pointer Dereference\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=223
Status	New

The variable declared in null at proxmark3/ecp.c in line 1355 is not initialized when it is used by X at proxmark3/ecp.c in line 752.

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	1412	773
Object	null	X

Code Snippet

File Name proxmark3/ecp.c

Method static int ecp_mul_comb(mbedtls_ecp_group *grp, mbedtls_ecp_point *R,

```
....
1412.       T = p_eq_g ? grp->T : NULL;
```

File Name proxmark3/ecp.c

Method static int ecp_normalize_jac(const mbedtls_ecp_group *grp, mbedtls_ecp_point *pt)

```
....
773.       MBEDTLS_MPI_CHK( mbedtls_mpi_mul_mpi( &pt->X,    &pt->X,
&ZZi    ) ); MOD_MUL( pt->X );
```

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=1050058&projectid=50048&pathid=7
Status	New

The readerAttack method calls the snprintf function, at line 1364 of proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1442	1442
Object	snprintf	snprintf

Code Snippet	
File Name	proxmark3/cmdhfmf.c
Method	void readerAttack(nonces_t ar_resp[], bool setEmulatorMem, bool doStandardAttack) {

```
....
1442.
      snprintf(cmd1,sizeof(cmd1),"%04x%08xFF078069%04x%08x",(uint32_t)
(sector_trailer[i].keyA>>32), (uint32_t) (sector_trailer[i].keyA
&0xFFFFFFFF),(uint32_t) (sector_trailer[i].keyB>>32), (uint32_t)
(sector_trailer[i].keyB &0xFFFFFFFF));
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=8
Status	New

The CmdHF14AMfELoad method calls the sprintf function, at line 1779 of proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1819	1819
Object	sprintf	sprintf

Code Snippet	
File Name	proxmark3/cmdhfmf.c

Method	int CmdHF14AMfELoad(const char *Cmd)

```
....
1819.        sprintf(fnameptr, ".eml");
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=9
Status	New

The CmdHF14AMfESave method calls the sprintf function, at line 1875 of proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1926	1926
Object	sprintf	sprintf

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfESave(const char *Cmd)

```
....
1926.                        sprintf(fnameptr, "%02X", buf[j]);
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=10
Status	New

The CmdHF14AMfESave method calls the sprintf function, at line 1875 of proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1933	1933
Object	sprintf	sprintf

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfESave(const char *Cmd)

```
....
1933.        sprintf(fnameptr, ".eml");
```

Unchecked Return Value\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=11
Status	New

The CmdHF14AMfCLoad method calls the sprintf function, at line 2258 of proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2316	2316
Object	sprintf	sprintf

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfCLoad(const char *Cmd)

```
....
2316.            sprintf(fnameptr, ".eml");
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=12
Status	New

The CmdHF14AMfCSave method calls the sprintf function, at line 2489 of proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2566	2566
Object	sprintf	sprintf

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfCSave(const char *Cmd) {

```
....
2566.                             sprintf(fnameptr, "%02x", buf[j]);
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=13
Status	New

The CmdHF14AMfCSave method calls the sprintf function, at line 2489 of proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2573	2573
Object	sprintf	sprintf

Code Snippet	
File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfCSave(const char *Cmd) {

```
....
2573.               sprintf(fnameptr, ".eml");
```

Unchecked Return Value\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=14
Status	New

The DecodeT55xxBlock method calls the snprintf function, at line 409 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	420	420
Object	snprintf	snprintf

Code Snippet	
File Name	proxmark3/cmdlft55xx.c
Method	bool DecodeT55xxBlock(){

```
....
420.                        snprintf(cmdStr, sizeof(buf),"%d %d",
bitRate[config.bitrate], config.inverted );
```

Unchecked Return Value\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=15
Status	New

The DecodeT55xxBlock method calls the snprintf function, at line 409 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	425	425
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method bool DecodeT55xxBlock(){

```
....
425.                        snprintf(cmdStr, sizeof(buf),"%d %d 8 5",
bitRate[config.bitrate], config.inverted );
```

Unchecked Return Value\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=16
Status	New

The DecodeT55xxBlock method calls the snprintf function, at line 409 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	430	430
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c

Method	bool DecodeT55xxBlock(){

```
....
430.                        snprintf(cmdStr, sizeof(buf),"%d %d 10 8",
bitRate[config.bitrate], config.inverted );
```

Unchecked Return Value\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=17
Status	New

The DecodeT55xxBlock method calls the snprintf function, at line 409 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	434	434
Object	snprintf	snprintf

Code Snippet	
File Name	proxmark3/cmdlft55xx.c
Method	bool DecodeT55xxBlock(){

```
....
434.                        snprintf(cmdStr, sizeof(buf),"%d %d 1",
bitRate[config.bitrate], config.inverted );
```

Unchecked Return Value\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=18
Status	New

The DecodeT55xxBlock method calls the snprintf function, at line 409 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	441	441
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	bool DecodeT55xxBlock(){

```
....
441.                        snprintf(cmdStr, sizeof(buf),"%d %d 6",
bitRate[config.bitrate], config.inverted );
```

Unchecked Return Value\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=19
Status	New

The DecodeT55xxBlock method calls the snprintf function, at line 409 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	451	451
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	bool DecodeT55xxBlock(){

```
....
451.                        snprintf(cmdStr, sizeof(buf),"%d 0 6",
bitRate[config.bitrate] );
```

Unchecked Return Value\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=20
Status	New

The DecodeT55xxBlock method calls the snprintf function, at line 409 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	458	458
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method bool DecodeT55xxBlock(){

```
....
458.                          snprintf(cmdStr, sizeof(buf),"%d %d 1",
bitRate[config.bitrate], config.inverted );
```

Unchecked Return Value\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=21
Status	New

The DecodeT55xxBlock method calls the snprintf function, at line 409 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	463	463
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method bool DecodeT55xxBlock(){

```
....
463.                          snprintf(cmdStr, sizeof(buf),"0 %d %d 1",
bitRate[config.bitrate], config.inverted );
```

Unchecked Return Value\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=22
Status	New

The T55xx_Print_DownlinkMode method calls the sprintf function, at line 479 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	482	482
Object	sprintf	sprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method void T55xx_Print_DownlinkMode (uint8_t downlink_mode)

```
....
482.        sprintf (Msg,"Downlink Mode used : ");
```

Unchecked Return Value\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=23
Status	New

The CmdT55xxWriteBlock method calls the snprintf function, at line 944 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1013	1013
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method int CmdT55xxWriteBlock(const char *Cmd) {

```
....
1013.       snprintf(pwdStr, sizeof(pwdStr), "pwd: 0x%08X", password);
```

Unchecked Return Value\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=24
Status	New

The GetBitRateStr method calls the snprintf function, at line 1299 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1304	1304
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetBitRateStr(uint32_t id, bool xmode) {

```
....
1304.                snprintf(retStr,sizeof(buf),"%d - RF/%d", id,
EM4x05_GET_BITRATE(id));
```

Unchecked Return Value\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=25
Status	New

The GetBitRateStr method calls the snprintf function, at line 1299 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1307	1307
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetBitRateStr(uint32_t id, bool xmode) {

```
....
1307.                       case 0:    snprintf(retStr,sizeof(buf),"%d -
RF/8",id);    break;
```

Unchecked Return Value\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=26
Status	New

The GetBitRateStr method calls the snprintf function, at line 1299 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1308	1308
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetBitRateStr(uint32_t id, bool xmode) {

```
....
1308.                   case 1:    snprintf(retStr,sizeof(buf),"%d -
RF/16",id);  break;
```

Unchecked Return Value\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=27
Status	New

The GetBitRateStr method calls the snprintf function, at line 1299 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1309	1309
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetBitRateStr(uint32_t id, bool xmode) {

```
....
1309.                   case 2:    snprintf(retStr,sizeof(buf),"%d -
RF/32",id);  break;
```

Unchecked Return Value\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=28
Status	New

The GetBitRateStr method calls the snprintf function, at line 1299 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1310	1310

Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetBitRateStr(uint32_t id, bool xmode) {

```
....
1310.                    case 3:   snprintf(retStr,sizeof(buf),"%d -
RF/40",id);  break;
```

Unchecked Return Value\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=29
Status	New

The GetBitRateStr method calls the snprintf function, at line 1299 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1311	1311
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetBitRateStr(uint32_t id, bool xmode) {

```
....
1311.                    case 4:   snprintf(retStr,sizeof(buf),"%d -
RF/50",id);  break;
```

Unchecked Return Value\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=30
Status	New

The GetBitRateStr method calls the snprintf function, at line 1299 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c

Line	1312	1312
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetBitRateStr(uint32_t id, bool xmode) {

```
....
1312.                  case 5:    snprintf(retStr,sizeof(buf),"%d -
RF/64",id);  break;
```

Unchecked Return Value\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=31
Status	New

The GetBitRateStr method calls the snprintf function, at line 1299 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1313	1313
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetBitRateStr(uint32_t id, bool xmode) {

```
....
1313.                  case 6:    snprintf(retStr,sizeof(buf),"%d -
RF/100",id); break;
```

Unchecked Return Value\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=32
Status	New

The GetBitRateStr method calls the snprintf function, at line 1299 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1314	1314
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetBitRateStr(uint32_t id, bool xmode) {

```
....
1314.                   case 7:    snprintf(retStr,sizeof(buf),"%d -
RF/128",id); break;
```

Unchecked Return Value\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=33
Status	New

The GetBitRateStr method calls the snprintf function, at line 1299 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1315	1315
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetBitRateStr(uint32_t id, bool xmode) {

```
....
1315.                   default:   snprintf(retStr,sizeof(buf),"%d -
(Unknown)",id); break;
```

Unchecked Return Value\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=34
Status	New

The GetSaferStr method calls the snprintf function, at line 1321 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1325	1325
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetSaferStr(uint32_t id) {

```
....
1325.        snprintf(retStr,sizeof(buf),"%d",id);
```

Unchecked Return Value\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=35
Status	New

The GetSaferStr method calls the snprintf function, at line 1321 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1327	1327
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetSaferStr(uint32_t id) {

```
....
1327.            snprintf(retStr,sizeof(buf),"%d - passwd",id);
```

Unchecked Return Value\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=36
Status	New

The GetSaferStr method calls the snprintf function, at line 1321 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1330	1330
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetSaferStr(uint32_t id) {

```
....
1330.               snprintf(retStr,sizeof(buf),"%d - testmode",id);
```

Unchecked Return Value\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=37
Status	New

The GetModulationStr method calls the snprintf function, at line 1336 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1341	1341
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetModulationStr(uint32_t id){

```
....
1341.               case 0: snprintf(retStr,sizeof(buf),"%d - DIRECT
(ASK/NRZ)",id); break;
```

Unchecked Return Value\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=38
Status	New

The GetModulationStr method calls the snprintf function, at line 1336 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1342	1342
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetModulationStr(uint32_t id){

```
....
1342.            case 1: snprintf(retStr,sizeof(buf),"%d - PSK 1 phase
change when input changes",id); break;
```

Unchecked Return Value\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=39
Status	New

The GetModulationStr method calls the snprintf function, at line 1336 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1343	1343
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetModulationStr(uint32_t id){

```
....
1343.            case 2:    snprintf(retStr,sizeof(buf),"%d - PSK 2
phase change on bitclk if input high",id); break;
```

Unchecked Return Value\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=40
Status	New

The GetModulationStr method calls the snprintf function, at line 1336 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1344	1344
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetModulationStr(uint32_t id){

```
....
1344.                 case 3: snprintf(retStr,sizeof(buf),"%d - PSK 3 phase
change on rising edge of input",id); break;
```

Unchecked Return Value\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=41
Status	New

The GetModulationStr method calls the snprintf function, at line 1336 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1345	1345
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetModulationStr(uint32_t id){

```
....
1345.                 case 4: snprintf(retStr,sizeof(buf),"%d - FSK 1 RF/8
RF/5",id); break;
```

Unchecked Return Value\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=42
Status	New

The GetModulationStr method calls the snprintf function, at line 1336 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1346	1346
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetModulationStr(uint32_t id){

```
....
1346.               case 5: snprintf(retStr,sizeof(buf),"%d - FSK 2 RF/8
RF/10",id); break;
```

Unchecked Return Value\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=43
Status	New

The GetModulationStr method calls the snprintf function, at line 1336 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1347	1347
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetModulationStr(uint32_t id){

```
....
1347.               case 6: snprintf(retStr,sizeof(buf),"%d - FSK 1a RF/5
RF/8",id); break;
```

Unchecked Return Value\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=44
Status	New

The GetModulationStr method calls the snprintf function, at line 1336 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1348	1348
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetModulationStr(uint32_t id){

```
....
1348.            case 7: snprintf(retStr,sizeof(buf),"%d - FSK 2a RF/10
RF/8",id); break;
```

Unchecked Return Value\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=45
Status	New

The GetModulationStr method calls the snprintf function, at line 1336 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1349	1349
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetModulationStr(uint32_t id){

```
....
1349.            case 8: snprintf(retStr,sizeof(buf),"%d -
Manchester",id); break;
```

Unchecked Return Value\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=46
Status	New

The GetModulationStr method calls the snprintf function, at line 1336 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1350	1350
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetModulationStr(uint32_t id){

```
....
1350.                case 16: snprintf(retStr,sizeof(buf),"%d -
Biphase",id); break;
```

Unchecked Return Value\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=47
Status	New

The GetModulationStr method calls the snprintf function, at line 1336 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1351	1351
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetModulationStr(uint32_t id){

```
....
1351.                case 0x18: snprintf(retStr,sizeof(buf),"%d - Biphase a
- AKA Conditional Dephase Encoding(CDP)",id); break;
```

Unchecked Return Value\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=48
Status	New

The GetModulationStr method calls the snprintf function, at line 1336 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1352	1352
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetModulationStr(uint32_t id){

```
....
1352.                case 17: snprintf(retStr,sizeof(buf),"%d -
Reserved",id); break;
```

Unchecked Return Value\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=49
Status	New

The GetModulationStr method calls the snprintf function, at line 1336 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1353	1353
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetModulationStr(uint32_t id){

```
....
1353.                default: snprintf(retStr,sizeof(buf),"0x%02X
(Unknown)",id); break;
```

Unchecked Return Value\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=50
Status	New

The GetModelStrFromCID method calls the snprintf function, at line 1358 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1363	1363
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetModelStrFromCID(uint32_t cid){

```
....
1363.        if (cid == 1) snprintf(retStr, sizeof(buf),"ATA5577M1");
```

Unchecked Return Value\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=51
Status	New

The GetModelStrFromCID method calls the snprintf function, at line 1358 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1364	1364
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetModelStrFromCID(uint32_t cid){

```
....
1364.        if (cid == 2) snprintf(retStr, sizeof(buf),"ATA5577M2");
```

Unchecked Return Value\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=52
Status	New

The GetSelectedModulationStr method calls the snprintf function, at line 1368 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1374	1374
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetSelectedModulationStr(uint8_t id){

```
....
1374.            case DEMOD_FSK:   snprintf(retStr,sizeof(buf),"FSK");
      break;
```

Unchecked Return Value\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=53
Status	New

The GetSelectedModulationStr method calls the snprintf function, at line 1368 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1375	1375
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetSelectedModulationStr(uint8_t id){

```
....
1375.            case DEMOD_FSK1: snprintf(retStr,sizeof(buf),"FSK1");
break;
```

Unchecked Return Value\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=54
Status	New

The GetSelectedModulationStr method calls the snprintf function, at line 1368 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1376	1376
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetSelectedModulationStr(uint8_t id){

```
....
1376.             case DEMOD_FSK1a:
snprintf(retStr,sizeof(buf),"FSK1a"); break;
```

Unchecked Return Value\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=55
Status	New

The GetSelectedModulationStr method calls the snprintf function, at line 1368 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1377	1377
Object	snprintf	snprintf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	char * GetSelectedModulationStr(uint8_t id){

```
....
1377.             case DEMOD_FSK2: snprintf(retStr,sizeof(buf),"FSK2");
break;
```

Unchecked Return Value\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=56
Status	New

The GetSelectedModulationStr method calls the snprintf function, at line 1368 of proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1378	1378
Object	snprintf	snprintf

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method char * GetSelectedModulationStr(uint8_t id){

```
....
1378.             case DEMOD_FSK2a:
snprintf(retStr,sizeof(buf),"FSK2a"); break;
```

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=1050058&projectid=50048&pathid=483
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1173	1173
Object	fgets	fgets

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfChk(const char *Cmd) {

```
....
1173.                         while (fgets(buf, sizeof(buf), f)) {
```

Improper Resource Access Authorization\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=484

Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1601	1601
Object	fgets	fgets

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfSim(const char *Cmd) {

```
....
1601.                        if (fgets(buf, sizeof(buf), f) == NULL) {
```

Improper Resource Access Authorization\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=485
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1832	1832
Object	fgets	fgets

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfELoad(const char *Cmd)

```
....
1832.                  if (fgets(buf, sizeof(buf), f) == NULL) {
```

Improper Resource Access Authorization\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=486
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2330	2330
Object	fgets	fgets

Code Snippet	
File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfCLoad(const char *Cmd)

```
....
2330.                    if (fgets(buf, sizeof(buf), f) == NULL) {
```

Improper Resource Access Authorization\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=487
Status	New

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1499	1499
Object	fgets	fgets

Code Snippet	
File Name	proxmark3/cmdlft55xx.c
Method	int CmdT55xxBruteForce(const char *Cmd) {

```
....
1499.               while( fgets(buf, sizeof(buf), f) ) {
```

Improper Resource Access Authorization\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=488
Status	New

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	817	817
Object	fgets	fgets

Code Snippet	
File Name	proxmark3/mifarehost.c
Method	int loadTraceCard(uint8_t *tuid) {

```
....
817.                if (fgets(buf, sizeof(buf), f) == NULL) {
```

Improper Resource Access Authorization\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=489
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1177	1177
Object	fgetc	fgetc

Code Snippet

File Name proxmark3/cmdhfmf.c

Method int CmdHF14AMfChk(const char *Cmd) {

```
....
1177.                          while (fgetc(f) != '\n' && !feof(f))
;  //goto next line
```

Improper Resource Access Authorization\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=490
Status	New

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1502	1502
Object	fgetc	fgetc

Code Snippet

File Name proxmark3/cmdlft55xx.c

Method int CmdT55xxBruteForce(const char *Cmd) {

```
....
1502.                while (fgetc(f) != '\n' && !feof(f)) ;  //goto
next line
```

Improper Resource Access Authorization\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=491
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1173	1173
Object	buf	buf

Code Snippet
File Name   proxmark3/cmdhfmf.c
Method   int CmdHF14AMfChk(const char *Cmd) {

```
....
1173.                        while (fgets(buf, sizeof(buf), f)) {
```

Improper Resource Access Authorization\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=492
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1601	1601
Object	buf	buf

Code Snippet
File Name   proxmark3/cmdhfmf.c
Method   int CmdHF14AMfSim(const char *Cmd) {

```
....
1601.                    if (fgets(buf, sizeof(buf), f) == NULL) {
```

Improper Resource Access Authorization\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=493
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1832	1832
Object	buf	buf

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfELoad(const char *Cmd)

```
....
1832.              if (fgets(buf, sizeof(buf), f) == NULL) {
```

Improper Resource Access Authorization\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=494
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2330	2330
Object	buf	buf

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfCLoad(const char *Cmd)

```
....
2330.                    if (fgets(buf, sizeof(buf), f) == NULL) {
```

Improper Resource Access Authorization\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=495
Status	New

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1499	1499
Object	buf	buf

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	int CmdT55xxBruteForce(const char *Cmd) {

```
....
1499.              while( fgets(buf, sizeof(buf), f) ) {
```

Improper Resource Access Authorization\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=496
Status	New

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	817	817
Object	buf	buf

Code Snippet
File Name proxmark3/mifarehost.c
Method int loadTraceCard(uint8_t *tuid) {

```
....
817.                if (fgets(buf, sizeof(buf), f) == NULL) {
```

Improper Resource Access Authorization\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=497
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	323	323
Object	keys	keys

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfDump(const char *Cmd)

```
....
323.                    size_t bytes_read = fread(keys[group][sectorNo], 
1, 6, fin);
```

Improper Resource Access Authorization\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=498
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	498	498
Object	keyA	keyA

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfRestore(const char *Cmd)

```
....
498.             size_t bytes_read = fread(keyA[sectorNo], 1, 6,
fkeys);
```

Improper Resource Access Authorization\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=499
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	507	507
Object	keyB	keyB

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfRestore(const char *Cmd)

```
....
507.             size_t bytes_read = fread(keyB[sectorNo], 1, 6,
fkeys);
```

Improper Resource Access Authorization\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=500
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	528	528
Object	bldata	bldata

Code Snippet	
File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfRestore(const char *Cmd)

```
....
528.                    size_t bytes_read = fread(bldata, 1, 16, fdump);
```

Improper Resource Access Authorization\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=501
Status	New

	Source	Destination
File	proxmark3/elite_crack.c	proxmark3/elite_crack.c
Line	562	562
Object	dump	dump

Code Snippet	
File Name	proxmark3/elite_crack.c
Method	int bruteforceFile(const char *filename, uint16_t keytable[])

```
....
562.        size_t bytes_read = fread(dump, 1, fsize, f);
```

Improper Resource Access Authorization\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=502
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	385	385
Object	Address	Address

Code Snippet	
File Name	proxmark3/hardnested_bruteforce.c
Method	static bool read_bench_data(statelist_t *test_candidates) {

```
....
385.        bytes_read = fread(&nonces_to_bruteforce, 1, 
sizeof(nonces_to_bruteforce), benchfile);
```

Improper Resource Access Authorization\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=503
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	391	391
Object	Address	Address

Code Snippet

File Name	proxmark3/hardnested_bruteforce.c
Method	static bool read_bench_data(statelist_t *test_candidates) {

```
....
391.                bytes_read = fread(&bf_test_nonce[i], 1,
sizeof(uint32_t), benchfile);
```

Improper Resource Access Authorization\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=504
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	397	397
Object	Address	Address

Code Snippet

File Name	proxmark3/hardnested_bruteforce.c
Method	static bool read_bench_data(statelist_t *test_candidates) {

```
....
397.                bytes_read = fread(&bf_test_nonce_par[i], 1,
sizeof(uint8_t), benchfile);
```

Improper Resource Access Authorization\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=505
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	403	403
Object	Address	Address

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static bool read_bench_data(statelist_t *test_candidates) {

```
....
403.         bytes_read = fread(&num_states, 1, sizeof(uint32_t), 
benchfile);
```

Improper Resource Access Authorization\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=506
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	409	409
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static bool read_bench_data(statelist_t *test_candidates) {

```
....
409.              bytes_read = fread(test_candidates->states[EVEN_STATE] 
+ states_read, 1, sizeof(uint32_t), benchfile);
```

Improper Resource Access Authorization\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=507
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	419	419
Object	Address	Address

Code Snippet

File Name	proxmark3/hardnested_bruteforce.c
Method	static bool read_bench_data(statelist_t *test_candidates) {

```
....
419.            bytes_read = fread(&temp, 1, sizeof(uint32_t),
benchfile);
```

Improper Resource Access Authorization\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=508
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	426	426
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name	proxmark3/hardnested_bruteforce.c
Method	static bool read_bench_data(statelist_t *test_candidates) {

```
....
426.            bytes_read = fread(test_candidates->states[ODD_STATE]
+ states_read, 1, sizeof(uint32_t), benchfile);
```

Improper Resource Access Authorization\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=509
Status	New

	Source	Destination
File	proxmark3/ikeys.c	proxmark3/ikeys.c
Line	745	745
Object	key	key

Code Snippet

File Name	proxmark3/ikeys.c
Method	int readKeyFile(uint8_t key[8])

```
....
745.         if (fread(key, sizeof(uint8_t), 8, f) == 8) {
```

Improper Resource Access Authorization\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=510
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1950	1950
Object	fprintf	fprintf

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfESave(const char *Cmd)

```
....
1950.                     fprintf(f, "%02X", buf[j]);
```

Improper Resource Access Authorization\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=511
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1951	1951
Object	fprintf	fprintf

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfESave(const char *Cmd)

```
....
1951.                fprintf(f,"\n");
```

Improper Resource Access Authorization\Path 30:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=512
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2597	2597
Object	fprintf	fprintf

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfCSave(const char *Cmd) {

```
....
2597.                           fprintf(f, "%02x", buf[j]);
```

Improper Resource Access Authorization\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=513
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2598	2598
Object	fprintf	fprintf

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfCSave(const char *Cmd) {

```
....
2598.                   fprintf(f,"\n");
```

Improper Resource Access Authorization\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=514
Status	New

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	787	787

Object	fprintf	fprintf

Code Snippet
File Name proxmark3/mifarehost.c
Method int saveTraceCard(void) {

```
....
787.                    fprintf(f, "%02x", *(traceCard + i * 16 + j));
```

Improper Resource Access Authorization\Path 33:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=515
Status New

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	789	789
Object	fprintf	fprintf

Code Snippet
File Name proxmark3/mifarehost.c
Method int saveTraceCard(void) {

```
....
789.                    fprintf(f,"\n");
```

Improper Resource Access Authorization\Path 34:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=516
Status New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	452	452
Object	fwrite	fwrite

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfDump(const char *Cmd)

```
....
452.                  fwrite(carddata, 1, 16*numblocks, fout);
```

Improper Resource Access Authorization\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=517
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	860	860
Object	fwrite	fwrite

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfNested(const char *Cmd) {

```
....
860.                                    fwrite ( tempkey, 1, 6, fkeys );
```

Improper Resource Access Authorization\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=518
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	863	863
Object	fwrite	fwrite

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfNested(const char *Cmd) {

```
....
863.                                    fwrite ( &standart, 1, 6, fkeys );
```

Improper Resource Access Authorization\Path 37:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=519
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	869	869
Object	fwrite	fwrite

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfNested(const char *Cmd) {

```
....
869.                                    fwrite ( tempkey, 1, 6, fkeys );
```

Improper Resource Access Authorization\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=520
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	872	872
Object	fwrite	fwrite

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfNested(const char *Cmd) {

```
....
872.                                    fwrite ( &standart, 1, 6, fkeys );
```

Improper Resource Access Authorization\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=521
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1350	1350

Object	fwrite	fwrite

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfChk(const char *Cmd) {

```
....
1350.                         fwrite(mkey, 1, 6, fkeys);
```

Improper Resource Access Authorization\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=522
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2060	2060
Object	fwrite	fwrite

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfEKeyPrn(const char *Cmd)

```
....
2060.                         fwrite(data, 1, 6, fkeys);
```

Improper Resource Access Authorization\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=523
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2067	2067
Object	fwrite	fwrite

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfEKeyPrn(const char *Cmd)

```
....
2067.                    fwrite(data+10, 1, 6, fkeys);
```

Improper Resource Access Authorization\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=524
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	269	269
Object	fwrite	fwrite

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static void write_benchfile(statelist_t *candidates) {

```
....
269.          fwrite(&nonces_to_bruteforce, 1,
sizeof(nonces_to_bruteforce), benchfile);
```

Improper Resource Access Authorization\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=525
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	271	271
Object	fwrite	fwrite

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static void write_benchfile(statelist_t *candidates) {

```
....
271.                fwrite(&(bf_test_nonce[i]), 1,
sizeof(bf_test_nonce[i]), benchfile);
```

Improper Resource Access Authorization\Path 44:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=526
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	272	272
Object	fwrite	fwrite

Code Snippet

File Name	proxmark3/hardnested_bruteforce.c
Method	static void write_benchfile(statelist_t *candidates) {

```
....
272.                    fwrite(&(bf_test_nonce_par[i]), 1,
sizeof(bf_test_nonce_par[i]), benchfile);
```

Improper Resource Access Authorization\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=527
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	275	275
Object	fwrite	fwrite

Code Snippet

File Name	proxmark3/hardnested_bruteforce.c
Method	static void write_benchfile(statelist_t *candidates) {

```
....
275.         fwrite(&num_states, 1, sizeof(num_states), benchfile);
```

Improper Resource Access Authorization\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=528
Status	New

	Source	Destination

File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	277	277
Object	fwrite	fwrite

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static void write_benchfile(statelist_t *candidates) {

```
....
277.                fwrite(&(candidates->states[EVEN_STATE][i]), 1,
sizeof(uint32_t), benchfile);
```

Improper Resource Access Authorization\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=529
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	280	280
Object	fwrite	fwrite

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static void write_benchfile(statelist_t *candidates) {

```
....
280.            fwrite(&num_states, 1, sizeof(num_states), benchfile);
```

Improper Resource Access Authorization\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=530
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	282	282
Object	fwrite	fwrite

Code Snippet
File Name proxmark3/hardnested_bruteforce.c

Method	static void write_benchfile(statelist_t *candidates) {
	`....`
	`282.                 fwrite(&(candidates->states[ODD_STATE][i]), 1, sizeof(uint32_t), benchfile);`

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=1050058&projectid=50048&pathid=531
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	315	315
Object	fin	fin

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfDump(const char *Cmd)

```
....
315.         if ((fin = fopen("dumpkeys.bin","rb")) == NULL) {
```

Incorrect Permission Assignment For Critical Resources\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=532
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	447	447
Object	fout	fout

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfDump(const char *Cmd)

```
....
447.                if ((fout = fopen("dumpdata.bin","wb")) == NULL) {
```

Incorrect Permission Assignment For Critical Resources\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=533
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	492	492
Object	fkeys	fkeys

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfRestore(const char *Cmd)

```
....
492.          if ((fkeys = fopen("dumpkeys.bin","rb")) == NULL) {
```

Incorrect Permission Assignment For Critical Resources\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=534
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	517	517
Object	fdump	fdump

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfRestore(const char *Cmd)

```
....
517.          if ((fdump = fopen("dumpdata.bin","rb")) == NULL) {
```

Incorrect Permission Assignment For Critical Resources\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=535
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	851	851
Object	fkeys	fkeys

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfNested(const char *Cmd) {

```
....
851.                    if ((fkeys = fopen("dumpkeys.bin","wb")) ==
NULL) {
```

Incorrect Permission Assignment For Critical Resources\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=536
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1172	1172
Object	f	f

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfChk(const char *Cmd) {

```
....
1172.                   if ((f = fopen( filename , "r"))) {
```

Incorrect Permission Assignment For Critical Resources\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=537
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1591	1591
Object	f	f

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfSim(const char *Cmd) {

```
....
1591.                 f = fopen(filename, "r");
```

Incorrect Permission Assignment For Critical Resources\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=538
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1822	1822
Object	f	f

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfELoad(const char *Cmd)

```
....
1822.         f = fopen(filename, "r");
```

Incorrect Permission Assignment For Critical Resources\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=539
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1936	1936
Object	f	f

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfESave(const char *Cmd)

```
....
1936.        f = fopen(filename, "w+");
```

Incorrect Permission Assignment For Critical Resources\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=540
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2050	2050
Object	fkeys	fkeys

Code Snippet	
File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfEKeyPrn(const char *Cmd)

```
....
2050.            if ((fkeys = fopen("dumpkeys.bin","wb")) == NULL) {
```

Incorrect Permission Assignment For Critical Resources\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=541
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2319	2319
Object	f	f

Code Snippet	
File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfCLoad(const char *Cmd)

```
....
2319.            f = fopen(filename, "r");
```

Incorrect Permission Assignment For Critical Resources\Path 12:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=542
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2576	2576
Object	f	f

Code Snippet

File Name proxmark3/cmdhfmf.c

Method int CmdHF14AMfCSave(const char *Cmd) {

```
....
2576.                 f = fopen(filename, "w+");
```

Incorrect Permission Assignment For Critical Resources\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=543
Status	New

	Source	Destination
File	proxmark3/ikeys.c	proxmark3/ikeys.c
Line	741	741
Object	f	f

Code Snippet

File Name proxmark3/ikeys.c

Method int readKeyFile(uint8_t key[8])

```
....
741.          f = fopen("iclass_key.bin", "rb");
```

Incorrect Permission Assignment For Critical Resources\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=544
Status	New

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c

Line	782	782
Object	f	f

Code Snippet

File Name	proxmark3/mifarehost.c
Method	int saveTraceCard(void) {

```
....
782.        f = fopen(traceFileName, "w+");
```

Incorrect Permission Assignment For Critical Resources\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=545
Status	New

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	809	809
Object	f	f

Code Snippet

File Name	proxmark3/mifarehost.c
Method	int loadTraceCard(uint8_t *tuid) {

```
....
809.        f = fopen(traceFileName, "r");
```

Incorrect Permission Assignment For Critical Resources\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=546
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1339	1339
Object	fkeys	fkeys

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfChk(const char *Cmd) {

```
....
1339.                FILE *fkeys = fopen("dumpkeys.bin","wb");
```

Incorrect Permission Assignment For Critical Resources\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=547
Status	New

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1491	1491
Object	f	f

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	int CmdT55xxBruteForce(const char *Cmd) {

```
....
1491.                FILE * f = fopen( filename , "r");
```

Incorrect Permission Assignment For Critical Resources\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=548
Status	New

	Source	Destination
File	proxmark3/elite_crack.c	proxmark3/elite_crack.c
Line	545	545
Object	f	f

Code Snippet

File Name	proxmark3/elite_crack.c
Method	int bruteforceFile(const char *filename, uint16_t keytable[])

```
....
545.         FILE *f = fopen(filename, "rb");
```

Incorrect Permission Assignment For Critical Resources\Path 19:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=549
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	268	268
Object	benchfile	benchfile

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static void write_benchfile(statelist_t *candidates) {

```
....
268.         FILE *benchfile = fopen(TEST_BENCH_FILENAME, "wb");
```

Incorrect Permission Assignment For Critical Resources\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=550
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	381	381
Object	benchfile	benchfile

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static bool read_bench_data(statelist_t *test_candidates) {

```
....
381.         FILE *benchfile = fopen(bench_file_path, "rb");
```

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=1050058&projectid=50048&pathid=551
Status	New

The CmdHF14AMfDump method in proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	315	315
Object	fopen	fopen

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfDump(const char *Cmd)

```
....
315.         if ((fin = fopen("dumpkeys.bin","rb")) == NULL) {
```

TOCTOU\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=552
Status	New

The CmdHF14AMfDump method in proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	447	447
Object	fopen	fopen

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfDump(const char *Cmd)

```
....
447.             if ((fout = fopen("dumpdata.bin","wb")) == NULL) {
```

TOCTOU\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=553
Status	New

The CmdHF14AMfRestore method in proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	492	492
Object	fopen	fopen

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfRestore(const char *Cmd)

```
....
492.        if ((fkeys = fopen("dumpkeys.bin","rb")) == NULL) {
```

TOCTOU\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=554
Status	New

The CmdHF14AMfRestore method in proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	517	517
Object	fopen	fopen

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfRestore(const char *Cmd)

```
....
517.        if ((fdump = fopen("dumpdata.bin","rb")) == NULL) {
```

TOCTOU\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=555
Status	New

The CmdHF14AMfNested method in proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	851	851
Object	fopen	fopen

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfNested(const char *Cmd) {

```
....
851.                    if ((fkeys = fopen("dumpkeys.bin","wb")) ==
NULL) {
```

TOCTOU\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=556
Status	New

The CmdHF14AMfChk method in proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1172	1172
Object	fopen	fopen

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfChk(const char *Cmd) {

```
....
1172.                   if ((f = fopen( filename , "r"))) {
```

TOCTOU\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=557
Status	New

The CmdHF14AMfChk method in proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1339	1339
Object	fopen	fopen

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfChk(const char *Cmd) {

```
....
1339.             FILE *fkeys = fopen("dumpkeys.bin","wb");
```

TOCTOU\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=558
Status	New

The CmdHF14AMfSim method in proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1591	1591
Object	fopen	fopen

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfSim(const char *Cmd) {

```
....
1591.             f = fopen(filename, "r");
```

TOCTOU\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=559
Status	New

The CmdHF14AMfELoad method in proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1822	1822
Object	fopen	fopen

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfELoad(const char *Cmd)

```
....
1822.        f = fopen(filename, "r");
```

TOCTOU\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=560
Status	New

The CmdHF14AMfESave method in proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1936	1936
Object	fopen	fopen

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfESave(const char *Cmd)

```
....
1936.        f = fopen(filename, "w+");
```

TOCTOU\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=561
Status	New

The CmdHF14AMfEKeyPrn method in proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2050	2050
Object	fopen	fopen

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfEKeyPrn(const char *Cmd)

```
....
2050.           if ((fkeys = fopen("dumpkeys.bin","wb")) == NULL) {
```

TOCTOU\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=562
Status	New

The CmdHF14AMfCLoad method in proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2319	2319
Object	fopen	fopen

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfCLoad(const char *Cmd)

```
....
2319.           f = fopen(filename, "r");
```

TOCTOU\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=563
Status	New

The CmdHF14AMfCSave method in proxmark3/cmdhfmf.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	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	2576	2576
Object	fopen	fopen

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfCSave(const char *Cmd) {

```
....
2576.                   f = fopen(filename, "w+");
```

TOCTOU\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=564
Status	New

The CmdT55xxBruteForce method in proxmark3/cmdlft55xx.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	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1491	1491
Object	fopen	fopen

Code Snippet
File Name proxmark3/cmdlft55xx.c
Method int CmdT55xxBruteForce(const char *Cmd) {

```
....
1491.                   FILE * f = fopen( filename , "r");
```

TOCTOU\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=565
Status	New

The bruteforceFile method in proxmark3/elite_crack.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	proxmark3/elite_crack.c	proxmark3/elite_crack.c
Line	545	545
Object	fopen	fopen

Code Snippet
File Name	proxmark3/elite_crack.c
Method	int bruteforceFile(const char *filename, uint16_t keytable[])

```
....
545.         FILE *f = fopen(filename, "rb");
```

TOCTOU\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=566
Status	New

The write_benchfile method in proxmark3/hardnested_bruteforce.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	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	268	268
Object	fopen	fopen

Code Snippet
File Name	proxmark3/hardnested_bruteforce.c
Method	static void write_benchfile(statelist_t *candidates) {

```
....
268.         FILE *benchfile = fopen(TEST_BENCH_FILENAME, "wb");
```

TOCTOU\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=567
Status	New

The read_bench_data method in proxmark3/hardnested_bruteforce.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	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	381	381
Object	fopen	fopen

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method static bool read_bench_data(statelist_t *test_candidates) {

```
....
381.        FILE *benchfile = fopen(bench_file_path, "rb");
```

TOCTOU\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=568
Status	New

The readKeyFile method in proxmark3/ikeys.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	proxmark3/ikeys.c	proxmark3/ikeys.c
Line	741	741
Object	fopen	fopen

Code Snippet
File Name proxmark3/ikeys.c
Method int readKeyFile(uint8_t key[8])

```
....
741.        f = fopen("iclass_key.bin", "rb");
```

TOCTOU\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=569
Status	New

The saveTraceCard method in proxmark3/mifarehost.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	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	782	782
Object	fopen	fopen

Code Snippet
File Name proxmark3/mifarehost.c
Method int saveTraceCard(void) {

```
....
782.        f = fopen(traceFileName, "w+");
```

TOCTOU\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=570
Status	New

The loadTraceCard method in proxmark3/mifarehost.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	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	809	809
Object	fopen	fopen

Code Snippet
File Name proxmark3/mifarehost.c
Method int loadTraceCard(uint8_t *tuid) {

```
....
809.        f = fopen(traceFileName, "r");
```

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=1050058&projectid=50048&pathid=273

Status	New

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	1400	1400
Object	i	i

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	uint32_t PackBits(uint8_t start, uint8_t len, uint8_t* bits){

```
....
1400.                tmp    |= bits[i] << j;
```

Unchecked Array Index\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=274
Status	New

	Source	Destination
File	proxmark3/cmdlfti.c	proxmark3/cmdlfti.c
Line	157	157
Object	maxPos	maxPos

Code Snippet

File Name	proxmark3/cmdlfti.c
Method	int CmdTIDemod(const char *Cmd)

```
....
157.     GraphBuffer[maxPos] = 800;
```

Unchecked Array Index\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=275
Status	New

	Source	Destination
File	proxmark3/cmdlfti.c	proxmark3/cmdlfti.c
Line	166	166
Object	maxPos	maxPos

Code Snippet	
File Name	proxmark3/cmdlfti.c
Method	int CmdTIDemod(const char *Cmd)

```
....
166.     GraphBuffer[maxPos] = 800;
```

Unchecked Array Index\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=276
Status	New

	Source	Destination
File	proxmark3/cmdlfti.c	proxmark3/cmdlfti.c
Line	206	206
Object	maxPos	maxPos

Code Snippet	
File Name	proxmark3/cmdlfti.c
Method	int CmdTIDemod(const char *Cmd)

```
....
206.       GraphBuffer[maxPos] = 800;
```

Unchecked Array Index\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=277
Status	New

	Source	Destination
File	proxmark3/crapto1.c	proxmark3/crapto1.c
Line	459	459
Object	size	size

Code Snippet	
File Name	proxmark3/crapto1.c
Method	uint32_t *lfsr_prefix_ks(uint8_t ks[8], int isodd)

```
....
459.         candidates[size] = -1;
```

Unchecked Array Index\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=278
Status	New

	Source	Destination
File	proxmark3/ecp.c	proxmark3/ecp.c
Line	201	201
Object	i	i

Code Snippet
File Name proxmark3/ecp.c
Method const mbedtls_ecp_group_id *mbedtls_ecp_grp_id_list(void)

```
....
201.          ecp_supported_grp_id[i] = MBEDTLS_ECP_DP_NONE;
```

Unchecked Array Index\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=279
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	200	200
Object	i	i

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method void prepare_bf_test_nonces(noncelist_t *nonces, uint8_t best_first_byte)

```
....
200.             bf_test_nonce[i] = test_nonce->nonce_enc;
```

Unchecked Array Index\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=280
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	201	201
Object	i	i

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method void prepare_bf_test_nonces(noncelist_t *nonces, uint8_t best_first_byte)

```
....
201.                bf_test_nonce_par[i] = test_nonce->par_enc;
```

Unchecked Array Index\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=281
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	202	202
Object	i	i

Code Snippet
File Name proxmark3/hardnested_bruteforce.c
Method void prepare_bf_test_nonces(noncelist_t *nonces, uint8_t best_first_byte)

```
....
202.                bf_test_nonce_2nd_byte[i] = (test_nonce->nonce_enc >>
16) & 0xff;
```

Unchecked Array Index\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=282
Status	New

	Source	Destination
File	proxmark3/hardnested_bruteforce.c	proxmark3/hardnested_bruteforce.c
Line	311	311
Object	bucket_count	bucket_count

Code Snippet

File Name	proxmark3/hardnested_bruteforce.c
Method	bool brute_force_bs(float *bf_rate, statelist_t *candidates, uint32_t cuid, uint32_t num_acquired_nonces, uint64_t maximum_states, noncelist_t *nonces, uint8_t *best_first_bytes)

```
....
311.                        buckets[bucket_count] = p;
```

Potential Off by One Error in Loops

Query Path:
CPP\Cx\CPP Heuristic\Potential Off by One Error in Loops Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection
NIST SP 800-53: SI-16 Memory Protection (P1)
OWASP Top 10 2017: A1-Injection

Description

Potential Off by One Error in Loops\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=89
Status	New

The buffer allocated by <= in proxmark3/cmdhfmf.c at line 280 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	321	321
Object	<=	<=

Code Snippet

File Name	proxmark3/cmdhfmf.c
Method	int CmdHF14AMfDump(const char *Cmd)

```
....
321.         for (int group=0; group<=1; group++) {
```

Potential Off by One Error in Loops\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=90
Status	New

The buffer allocated by <= in proxmark3/cmdhfmf.c at line 280 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	383	383
Object	<=	<=

Code Snippet
File Name proxmark3/cmdhfmf.c
Method int CmdHF14AMfDump(const char *Cmd)

```
....
383.                                    for (int k=0; k<=1; k++) {
```

Potential Off by One Error in Loops\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=91
Status	New

The buffer allocated by <= in proxmark3/crapto1.c at line 53 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	proxmark3/crapto1.c	proxmark3/crapto1.c
Line	68	68
Object	<=	<=

Code Snippet
File Name proxmark3/crapto1.c
Method static void bucket_sort_intersect(uint32_t* const estart, uint32_t* const estop,

```
....
68.             for (uint32_t j = 0x00; j <= 0xff; j++) {
```

Potential Off by One Error in Loops\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=92
Status	New

The buffer allocated by <= in proxmark3/crapto1.c at line 53 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	proxmark3/crapto1.c	proxmark3/crapto1.c
Line	88	88
Object	<=	<=

Code Snippet
File Name proxmark3/crapto1.c
Method static void bucket_sort_intersect(uint32_t* const estart, uint32_t* const estop,

```
....
88.         for (uint32_t j = 0x00; j <= 0xff; j++) {
```

Potential Off by One Error in Loops\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=93
Status	New

The buffer allocated by <= in proxmark3/crapto1.c at line 218 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	proxmark3/crapto1.c	proxmark3/crapto1.c
Line	243	243
Object	<=	<=

Code Snippet
File Name proxmark3/crapto1.c
Method struct Crypto1State* lfsr_recovery32(uint32_t ks2, uint32_t in)

```
....
243.             for (uint32_t j = 0; j <= 0xff; j++) {
```

Potential Off by One Error in Loops\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=94
Status	New

The buffer allocated by <= in proxmark3/crapto1.c at line 218 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	proxmark3/crapto1.c	proxmark3/crapto1.c

Line	271	271
Object	<=	<=

Code Snippet
File Name proxmark3/crapto1.c
Method struct Crypto1State* lfsr_recovery32(uint32_t ks2, uint32_t in)

```
....
271.                  for (uint32_t j = 0; j <= 0xff; j++)
```

Potential Off by One Error in Loops\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=95
Status	New

The buffer allocated by <= in proxmark3/lfdemod.c at line 979 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	1054	1054
Object	<=	<=

Code Snippet
File Name proxmark3/lfdemod.c
Method bool DetectST(uint8_t buffer[], size_t *size, int *foundclock, size_t *ststart, size_t *stend) {

```
....
1054.                 for ( i=0; i <= (clk/4); ++i ) {
```

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=1050058&projectid=50048&pathid=266
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c

Line	1373	1373
Object	sector_trailer	sizeof

Code Snippet
File Name proxmark3/cmdhfmf.c
Method void readerAttack(nonces_t ar_resp[], bool setEmulatorMem, bool doStandardAttack) {

```
....
1373.        memset(sector_trailer, 0x00, sizeof(sector_trailer));
```

Sizeof Pointer Argument\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=267
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1376	1376
Object	stSector	sizeof

Code Snippet
File Name proxmark3/cmdhfmf.c
Method void readerAttack(nonces_t ar_resp[], bool setEmulatorMem, bool doStandardAttack) {

```
....
1376.        memset(stSector, 0x00, sizeof(stSector));
```

Sizeof Pointer Argument\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=268
Status	New

	Source	Destination
File	proxmark3/cmdhfmf.c	proxmark3/cmdhfmf.c
Line	1378	1378
Object	key_cnt	sizeof

Code Snippet
File Name proxmark3/cmdhfmf.c

Method	void readerAttack(nonces_t ar_resp[], bool setEmulatorMem, bool doStandardAttack) {

```
....
1378.        memset(key_cnt, 0x00, sizeof(key_cnt));
```

Sizeof Pointer Argument\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=269
Status	New

	Source	Destination
File	proxmark3/lfdemod.c	proxmark3/lfdemod.c
Line	993	993
Object	waveLen	sizeof

Code Snippet

File Name	proxmark3/lfdemod.c
Method	bool DetectST(uint8_t buffer[], size_t *size, int *foundclock, size_t *ststart, size_t *stend) {

```
....
993.        memset(waveLen, 0, sizeof(waveLen));
```

Sizeof Pointer Argument\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=270
Status	New

	Source	Destination
File	proxmark3/mifareutil.c	proxmark3/mifareutil.c
Line	105	105
Object	dcmd	sizeof

Code Snippet

File Name	proxmark3/mifareutil.c
Method	int mifare_sendcmd_short(struct Crypto1State *pcs, uint8_t crypted, uint8_t cmd, uint8_t data, uint8_t *answer, uint8_t *answer_parity, uint32_t *timing) {

```
....
105.        memcpy(ecmd, dcmd, sizeof(dcmd));
```

Sizeof Pointer Argument\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=271
Status	New

	Source	Destination
File	proxmark3/mifareutil.c	proxmark3/mifareutil.c
Line	114	114
Object	ecmd	sizeof

Code Snippet

File Name proxmark3/mifareutil.c

Method int mifare_sendcmd_short(struct Crypto1State *pcs, uint8_t crypted, uint8_t cmd, uint8_t data, uint8_t *answer, uint8_t *answer_parity, uint32_t *timing) {

```
....
114.            ReaderTransmitPar(ecmd, sizeof(ecmd), par, timing);
```

Sizeof Pointer Argument\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=272
Status	New

	Source	Destination
File	proxmark3/mifareutil.c	proxmark3/mifareutil.c
Line	116	116
Object	dcmd	sizeof

Code Snippet

File Name proxmark3/mifareutil.c

Method int mifare_sendcmd_short(struct Crypto1State *pcs, uint8_t crypted, uint8_t cmd, uint8_t data, uint8_t *answer, uint8_t *answer_parity, uint32_t *timing) {

```
....
116.            ReaderTransmit(dcmd, sizeof(dcmd), timing);
```

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=1050058&projectid=50048&pathid=75
Status	New

	Source	Destination
File	proxmark3/lgc.c	proxmark3/lgc.c
Line	471	471
Object	sizeof	sizeof

Code Snippet

File Name	proxmark3/lgc.c
Method	static lu_mem traversetable (global_State *g, Table *h) {

```
....
471.                                 sizeof(Proto *) * f->sizep +
```

Use of Sizeof On a Pointer Type\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=76
Status	New

	Source	Destination
File	proxmark3/lgc.c	proxmark3/lgc.c
Line	1038	1038
Object	sizeof	sizeof

Code Snippet

File Name	proxmark3/lgc.c
Method	static lu_mem singlestep (lua_State *L) {

```
....
1038.        g->GCmemtrav = g->strt.size * sizeof(GCObject*);
```

Use of Sizeof On a Pointer Type\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=77
Status	New

	Source	Destination
File	proxmark3/lobject.c	proxmark3/lobject.c

Line	208	208
Object	sizeof	sizeof

Code Snippet
File Name proxmark3/lobject.c
Method const char *luaO_pushvfstring (lua_State *L, const char *fmt, va_list argp) {

```
....
208.            char buff[4*sizeof(void *) + 8]; /* should be enough space for a `%p' */
```

Use of Sizeof On a Pointer Type\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=78
Status	New

	Source	Destination
File	proxmark3/rsa_internal.c	proxmark3/rsa_internal.c
Line	94	94
Object	sizeof	sizeof

Code Snippet
File Name proxmark3/rsa_internal.c
Method int mbedtls_rsa_deduce_primes(mbedtls_mpi const *N,

```
....
94.      const size_t num_primes = sizeof( primes ) / sizeof( *primes );
```

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=1050058&projectid=50048&pathid=263
Status	New

	Source	Destination

File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	657	657
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name proxmark3/mifarehost.c
Method int mfCWipe(uint32_t numSectors, bool gen1b, bool wantWipe, bool wantFill) {

```
....
657.        uint8_t cmdParams = wantWipe + wantFill * 0x02 + gen1b *
0x04;
```

Arithmenic Operation On Boolean\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=264
Status	New

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	657	657
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name proxmark3/mifarehost.c
Method int mfCWipe(uint32_t numSectors, bool gen1b, bool wantWipe, bool wantFill) {

```
....
657.        uint8_t cmdParams = wantWipe + wantFill * 0x02 + gen1b *
0x04;
```

Arithmenic Operation On Boolean\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=265
Status	New

	Source	Destination
File	proxmark3/mifarehost.c	proxmark3/mifarehost.c
Line	657	657
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name	proxmark3/mifarehost.c
Method	int mfCWipe(uint32_t numSectors, bool gen1b, bool wantWipe, bool wantFill) {

```
....
657.        uint8_t cmdParams = wantWipe + wantFill * 0x02 + gen1b *
0x04;
```

Information Exposure Through Comments

Query Path:
CPP\Cx\CPP Low Visibility\Information Exposure Through Comments Version:1

Categories

FISMA 2014: Identification And Authentication
NIST SP 800-53: SC-28 Protection of Information at Rest (P1)

Description

Information Exposure Through Comments\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=571
Status	New

	Source	Destination
File	proxmark3/cmdlft55xx.c	proxmark3/cmdlft55xx.c
Line	803	803
Object	PWD =	PWD =

Code Snippet

File Name	proxmark3/cmdlft55xx.c
Method	//uint8_t PWD = PackBits(si, 1, DemodBuffer); si += 1;

```
....
803.            //uint8_t PWD       = PackBits(si, 1, DemodBuffer); si
+= 1;
```

Information Exposure Through Comments\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050058&projectid=50048&pathid=572
Status	New

	Source	Destination
File	proxmark3/mifareutil.c	proxmark3/mifareutil.c
Line	199	199
Object	cipher (N	cipher (N

Code Snippet

File Name	proxmark3/mifareutil.c
Method	// Generate (encrypted) nr+parity by loading it into the cipher (Nr)

```
....
        // Generate (encrypted) nr+parity by loading it into the
cipher (Nr)
```

Buffer Overflow IndexFromInput

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

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 AddressOfLocalVarReturned

Risk

What might happen

A use after free error will cause code to use an area of memory previously assigned with a specific value, which has since been freed and may have been overwritten by another value. This error will likely cause unexpected behavior, memory corruption and crash errors. In some cases where the freed and used section of memory is used to determine execution flow, and the error can be induced by an attacker, this may result in execution of malicious code.

Cause

How does it happen

Pointers to variables allow code to have an address with a set size to a dynamically allocated variable. Eventually, the pointer's destination may become free - either explicitly in code, such as when programmatically freeing this variable, or implicitly, such as when a local variable is returned - once it is returned, the variable's scope is released. Once freed, this memory will be re-used by the application, overwritten with new data. At this point, dereferencing this pointer will potentially resolve newly written and unexpected data.

General Recommendations

How to avoid it

- Do not return local variables or pointers
- Review code to ensure no flow allows use of a pointer after it has been explicitly freed

Source Code Examples

CPP

Use of Variable after It was Freed

```
free(input);
printf("%s", input);
```

Use of Pointer to Local Variable That Was Freed On Return

```
int* func1()
{
      int i;
      i = 1;
      return &i;
}

void func2()
```

```
{
      int j;
      j = 5;
}

//..
      int * i = func1();
      printf("%d\r\n", *i); // Output could be 1 or Segmentation Fault
      func2();
      printf("%d\r\n", *i); // Output is 5, which is j's value, as func2() overwrote data in
the stack
//..
```

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

MemoryFree on StackVariable

Risk

What might happen

Undefined Behavior may result with a crash. Crashes may give an attacker valuable information about the system and the program internals. Furthermore, it may leave unprotected files (e.g memory) that may be exploited.

Cause

How does it happen

Calling free() on a variable that was not dynamically allocated (e.g. malloc) will result with an Undefined Behavior.

General Recommendations

How to avoid it

Use free() only on dynamically allocated variables in order to prevent unexpected behavior from the compiler.

Source Code Examples

CPP

Bad - Calling free() on a static variable

```
void clean_up(){
  char temp[256];
  do_something();
  free(tmp);
  return;
}
```

Good - Calling free() only on variables that were dynamically allocated

```
void clean_up(){
  char *buff;
  buff = (char*) malloc(1024);
  free(buff);
  return;
}
```

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

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);
```

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

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;
}
```

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

```
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

```
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

```
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

```
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

```c
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

Heap Inspection

Risk

What might happen

All variables stored by the application in unencrypted memory can potentially be retrieved by an unauthorized user, with privlieged access to the machine. For example, a privileged attacker could attach a debugger to the running process, or retrieve the process's memory from the swapfile or crash dump file.
Once the attacker finds the user passwords in memory, these can be reused to easily impersonate the user to the system.

Cause

How does it happen

String variables are immutable - in other words, once a string variable is assigned, its value cannot be changed or removed. Thus, these strings may remain around in memory, possibly in multiple locations, for an indefinite period of time until the garbage collector happens to remove it. Sensitive data, such as passwords, will remain exposed in memory as plaintext with no control over their lifetime.

General Recommendations

How to avoid it

Generic Guidance:

- Do not store senstiive data, such as passwords or encryption keys, in memory in plaintext, even for a short period of time.
- Prefer to use specialized classes that store encrypted memory.
- Alternatively, store secrets temporarily in mutable data types, such as byte arrays, and then promptly zeroize the memory locations.

Specific Recommendations - Java:

- Instead of storing passwords in immutable strings, prefer to use an encrypted memory object, such as SealedObject.

Specific Recommendations - .NET:

- Instead of storing passwords in immutable strings, prefer to use an encrypted memory object, such as SecureString or ProtectedData.

Source Code Examples

Java

Plaintext Password in Immutable String

```
class Heap_Inspection
{
  private string password;

  void setPassword()
  {
      password = System.console().readLine("Enter your password: ");
   }
}
```

Password Protected in Memory

```
class Heap_Inspection_Fixed
{
  private SealedObject password;

  void setPassword()
 {
      byte[] sKey = getKeyFromConfig();
      Cipher c = Cipher.getInstance("AES");
      c.init(Cipher.ENCRYPT_MODE, sKey);

      char[] input = System.console().readPassword("Enter your password: ");
      password = new SealedObject(Arrays.asList(input), c);

    //Zero out the possible password, for security.
    Arrays.fill(password, '0');
  }
}
```

CPP

Vulnerable C code

```
/* Vulnerable to heap inspection */

#include <stdio.h>


void somefunc(){
      printf("Yea, I'm just being called for the heap of it..\n");
}

void authfunc(){
        char* password = (char *) malloc(256);
        char ch;
        ssize_t k;
            int i=0;
        while(k = read(0, &ch, 1) > 0)
        {
                if (ch == '\n'){
                        password[i]='\0';
                        break;
                } else{
                        password[i++]=ch;
                        fflush(0);
                }
        }
        printf("Password: %s\n",&password[0]);
}

int main()
{
    printf("Please enter a password:\n");

      authfunc();
      printf("You can now dump memory to find this password!");
      somefunc();
```

```
        gets();

}
```

Safe C code

```
/* Pesumably safe heap */

#include <stdio.h>
#include <string.h>

#define STDIN_FILENO 0

void somefunc(){
        printf("Yea, I'm just being called for the heap of it..\n");
}

void authfunc(){
        char* password = (char*) malloc(256);
        int i=0;
        char ch;
        ssize_t k;
        while(k = read(STDIN_FILENO, &ch, 1) > 0)
        {
                if (ch == '\n'){
                        password[i]='\0';
                        break;
                } else{
                        password[i++]=ch;
                        fflush(0);
                }
        }
        i=0;
        memset(password,'\0',256);
}

int main()

{

        printf("Please enter a password:\n");
        authfunc();
        somefunc();
        char ch;
        while(read(STDIN_FILENO, &ch, 1) > 0)
        {
                if (ch == '\n')
                        break;
        }
}
```

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 Variable

Weakness ID: 457 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code uses a variable that has not been initialized, leading to unpredictable or unintended results.

Extended Description

In some languages, such as C, an uninitialized variable contains contents of previously-used memory. An attacker can sometimes control or read these contents.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C: *(Sometimes)*

C++: *(Sometimes)*

Perl: *(Often)*

All

Common Consequences

Scope	Effect
Availability Integrity	Initial variables usually contain junk, which can not be trusted for consistency. This can lead to denial of service conditions, or modify control flow in unexpected ways. In some cases, an attacker can "pre-initialize" the variable using previous actions, which might enable code execution. This can cause a race condition if a lock variable check passes when it should not.
Authorization	Strings that are not initialized are especially dangerous, since many functions expect a null at the end -- and only at the end -- of a string.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

The following switch statement is intended to set the values of the variables aN and bN, but in the default case, the programmer has accidentally set the value of aN twice. As a result, bN will have an undefined value.

(Bad Code)

Example Language: **C**

```
switch (ctl) {
case -1:
aN = 0;
bN = 0;
break;
case 0:
aN = i;
bN = -i;
break;
case 1:
aN = i + NEXT_SZ;
bN = i - NEXT_SZ;
break;
default:
```

```
aN = -1;
aN = -1;
break;
}
repaint(aN, bN);
```

Most uninitialized variable issues result in general software reliability problems, but if attackers can intentionally trigger the use of an uninitialized variable, they might be able to launch a denial of service attack by crashing the program. Under the right circumstances, an attacker may be able to control the value of an uninitialized variable by affecting the values on the stack prior to the invocation of the function.

Example 2

Example Languages: **C++ and Java**

```
int foo;
void bar() {
if (foo==0)
/.../
/../
}
```

Observed Examples

Reference	Description
CVE-2008-0081	Uninitialized variable leads to code execution in popular desktop application.
CVE-2007-4682	Crafted input triggers dereference of an uninitialized object pointer.
CVE-2007-3468	Crafted audio file triggers crash when an uninitialized variable is used.
CVE-2007-2728	Uninitialized random seed variable used.

Potential Mitigations

Phase: Implementation

Assign all variables to an initial value.

Phase: Build and Compilation

Most compilers will complain about the use of uninitialized variables if warnings are turned on.

Phase: Requirements

The choice could be made to use a language that is not susceptible to these issues.

Phase: Architecture and Design

Mitigating technologies such as safe string libraries and container abstractions could be introduced.

Other Notes

Before variables are initialized, they generally contain junk data of what was left in the memory that the variable takes up. This data is very rarely useful, and it is generally advised to pre-initialize variables or set them to their first values early. If one forgets -- in the C language -- to initialize, for example a char *, many of the simple string libraries may often return incorrect results as they expect the null termination to be at the end of a string.

Stack variables in C and C++ are not initialized by default. Their initial values are determined by whatever happens to be in their location on the stack at the time the function is invoked. Programs should never use the value of an uninitialized variable. It is not uncommon for programmers to use an uninitialized variable in code that handles errors or other rare and exceptional circumstances. Uninitialized variable warnings can sometimes indicate the presence of a typographic error in the code.

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	Weakness Base	456	Missing Initialization	**Development Concepts (primary)699 Research Concepts**

				(primary)1000
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Uninitialized variable
7 Pernicious Kingdoms			Uninitialized Variable

White Box Definitions

A weakness where the code path has:

1. start statement that defines variable
2. end statement that accesses the variable
3. the code path does not contain a statement that assigns value to the variable

References

mercy. "Exploiting Uninitialized Data". Jan 2006. < http://www.felinemenace.org/~mercy/papers/UBehavior/UBehavior.zip>.

Microsoft Security Vulnerability Research & Defense. "MS08-014 : The Case of the Uninitialized Stack Variable Vulnerability". 2008-03-11. <http://blogs.technet.com/swi/archive/2008/03/11/the-case-of-the-uninitialized-stack-variable-vulnerability.aspx>.

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, Description, Relationships, Observed Example, Other Notes, References, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences, Demonstrative Examples, Potential Mitigations		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Uninitialized Variable

BACK TO TOP

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());
```

Stored Buffer Overflow boundcpy

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));
    }
}
```

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';
```

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

Potential Off by One Error in Loops

Risk

What might happen

An off by one error may result in overwriting or over-reading of unintended memory; in most cases, this can result in unexpected behavior and even application crashes. In other cases, where allocation can be controlled by an attacker, a combination of variable assignment and an off by one error can result in execution of malicious code.

Cause

How does it happen

Often when designating variables to memory, a calculation error may occur when determining size or length that is off by one.

For example in loops, when allocating an array of size 2, its cells are counted as 0,1 - therefore, if a For loop iterator on the array is incorrectly set with the start condition i=0 and the continuation condition i<=2, three cells will be accessed instead of 2, and an attempt will be made to write or read cell [2], which was not originally allocated, resulting in potential corruption of memory outside the bounds of the originally assigned array.

Another example occurs when a null-byte terminated string, in the form of a character array, is copied without its terminating null-byte. Without the null-byte, the string representation is unterminated, resulting in certain functions to over-read memory as they expect the missing null terminator.

General Recommendations

How to avoid it

- Always ensure that a given iteration boundary is correct:
    - With array iterations, consider that arrays begin with cell 0 and end with cell n-1, for a size n array.
    - With character arrays and null-byte terminated string representations, consider that the null byte is required and should not be overwritten or ignored; ensure functions in use are not vulnerable to off-by-one, specifically for instances where null-bytes are automatically appended after the buffer, instead of in place of its last character.
- Where possible, use safe functions that manage memory and are not prone to off-by-one errors.

Source Code Examples

CPP

Off-By-One in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i <= 5; i++)
{
      ptr[i] = i * 2 + 1; // ptr[5] will be set, but is out of bounds
}
```

Proper Iteration in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1; // ptr[0-4] are well defined
}
```

Off-By-One in strncat

```
strncat(buf, input, sizeof(buf) - strlen(buf)); // actual value should be sizeof(buf)-
strlen(buf)-1 - this form will overwrite the terminating nullbyte
```

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	Cigital	External
	updated Time of Introduction		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Description, Relationships, Taxonomy Mappings		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Relationships		

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 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

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

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--;
               }
            }
        }
```

Information Leak Through Comments

Weakness ID: 615 *(Weakness Variant)* **Status:** Incomplete

Description

Description Summary

While adding general comments is very useful, some programmers tend to leave important data, such as: filenames related to the web application, old links or links which were not meant to be browsed by users, old code fragments, etc.

Extended Description

An attacker who finds these comments can map the application's structure and files, expose hidden parts of the site, and study the fragments of code to reverse engineer the application, which may help develop further attacks against the site.

Time of Introduction

- Implementation

Demonstrative Examples

Example 1

The following comment, embedded in a JSP, will be displayed in the resulting HTML output.

(Bad Code)

Example Languages: **HTML and JSP**

```
<!-- FIXME: calling this with more than 30 args kills the JDBC server -->
```

Observed Examples

Reference	Description
CVE-2007-6197	Version numbers and internal hostnames leaked in HTML comments.
CVE-2007-4072	CMS places full pathname of server in HTML comment.
CVE-2009-2431	blog software leaks real username in HTML comment.

Potential Mitigations

Remove comments which have sensitive information about the design/implementation of the application. Some of the comments may be exposed to the user and affect the security posture of the application.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Variant	540	Information Leak Through Source Code	**Development Concepts (primary)699** **Research Concepts (primary)1000**

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	Anonymous Tool Vendor (under NDA)		Externally Mined

Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Sean Eidemiller	Cigital	External
	added/updated demonstrative examples		
2008-07-01	Eric Dalci	Cigital	External
	updated Potential Mitigations, Time of Introduction		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2008-10-14	CWE Content Team	MITRE	Internal
	updated Description		
2009-03-10	CWE Content Team	MITRE	Internal

	updated Demonstrative Examples		
2009-07-27	CWE Content Team	MITRE	Internal
	updated Observed Examples, Taxonomy Mappings		

BACK TO TOP

Scanned Languages

Language	Hash Number	Change Date
CPP	4541647240435660	6/19/2024
Common	0105849645654507	6/19/2024