NEWS

X41 D-SEC GmbH Advisory: X41-2019-008

Vulnerable Components in Cerner medico

Severity Rating:
Medium

Confirmed Affected Versions:
unknown

Confirmed Patched Versions:
Defect 226386, Hotfix H26001200000

Vendor:
Cerner Health Services Deutschland GmbH

Vendor URL:
https://www.cerner.com

Vendor Reference:
https://www.cerner.com/de/de/loesungen/medico

Vector:
Adjacent Network

Credit:
X41 D-SEC GmbH

Status:
Public

CVE:
CVE-2020-11674, CVE-2020-11675, CVE-2020-11676, CVE-2020-11677

CVSS Score:
6.3

CVSS Vector:
CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L

Advisory-URL:
None

Summary and Impact

During a penetration test, a Cerner medico hospital information management system was discovered with numerous security issues. A process called `slim_proxy` was observed to be running and listening on the network. Upon investigating this custom component, its source code was discovered on the system. Copyright banners revealed that this code is custom made by Cerner. During a cursory investigation, numerous security issues were discovered which, if reachable by an attacker, would allow the attacker to take complete control of the software.
The folder that was copied during the investigation does not contain all components of the software. Given the very limited view, it cannot be estimated whether the vulnerabilties are reachable from the network. However, the number of potentially critical vulnerabilities in a very sensitive information system prompted X41 to issue this advisory.

Product Description

The Cerner medico hospital information management system helps manage and control electronic health records in hospitals.

Analysis

As part of the pentest, X41 copied a folder called `BINC` containing the source files `crypt.cpp`, `dpscrypt.c`, `mkdict.c`, `ntmkcr.c`, and `slim_proxy.c`, as well as compiled binaries or object files. Most of the files have copyright headers attributing Cerner. Since this is not the complete software, no thorough audit was performed, and no documentation available, the relation between the tools is only partially known. Bugs in the argument parsing may only be exploitable by a local user, or may be triggerable through the network if called by another tool on the system. In either case, such bugs are considered to be indicative of the kinds of mistakes the developers made.

`slim_proxy.c` contains a function `showInode()` which copies the result of `iso2utf()` into a buffer of 500 bytes. Since the result of `iso2utf()` can potentially be 1000 bytes long, this could result in a buffer overflow. The function `refreshList()` uses the variable output without initialization, potentially leading to data corruption.

`dpscrypt.c` also contains multiple buffer overflow vulnerabilities. Command line arguments as well as environment variables are copied into a fixed-length buffer using `strcpy()`. Since no length is given, the value may overflow the buffer. Furthermore, the purpose of this utility appears to be to encrypt passwords with the broken DES encryption algorithm. The file `crypt.cpp` contains an implementation of the broken DES encryption algorithm to facilitate this.

`ntmkcr.c` uses the variable `opt_string` as both source and destination of `sprintf()` in the function `mod_key_opts()`, resulting in undefined behaviour and potential data corruption.

`mkdict.c` has a similar bug where the variable `sbuf` is used as both source and destination of `sprintf()` in the function `mask_tick()`. It also contains a buffer overflow by copying an environment variable into a 30-byte buffer using `strcpy()` without any boundary checks. Data is not escaped when writing a certain format output file, for example in the function `write_csvfile()`, where values are written without escaping the field delimiter (semicolon). If any of the fields would contain a semicolon, it would break out of the field.
The code quality overall is considered poor and should not be used to handle data of which the confidentiality or integrity is important. Finally, mitigations such as stack canaries, FORTIFY_SOURCE, RELRO, and PIE are not enabled in the compiled binaries.

Workarounds

Apply the patches to have the state of 2020-04-01.

Timeline

2019-11-12
Issue found

2020-01-20
Customer of X41 grants permission to pass the advisory

2020-01-20
BSI contacted by X41

2020-01-22
BSI approved to take care of contacting the vendor and to notify affected hospitals

2020-02-04
Vendor released Defect 226385, Hotfix H26001102000 to mitigate issues in slim_proxy.c

2020-02-25
Conference with Cerner BSI and X41

2020-02-26
Vendor released Defect 226386, Hotfix H26001200000

2020-04-01
Vendor wrote in a statement regarding CVE registration that all risks have been remediated

2020-04-06
X41 sent preliminary advisory to BSI with request to forward it to the vendor

2020-04-07
Vendor received preliminary advisory from BSI

2020-04-21
Vendor sent version numbers of the mitigations to BSI

2020-04-23
X41 released advisory

About X41 D-SEC GmbH

X41 is an expert provider for application security services. Having extensive industry experience and expertise in the area of information security, a strong core security team of world class security experts enables X41 to perform premium security services.

Fields of expertise in the area of application security are security centered code reviews, binary reverse engineering and vulnerability discovery. Custom research and IT security consulting and support services are core competencies of X41.

Author: Luc Gommans
Date: April 23, 2020

«
Pro-bono Pentests for COVID-19-related Apps & Software
»
Vulnerabilities and Coordinated Disclosure

CONTACT

X41 D-SEC GmbH
Krefelder Str. 123
52070 Aachen

+49 (0) 241 9809418-0
+49 (0) 241 9809418-9
info@x41-dsec.de

PGP Key

CONNECT