# CWE Detail – CWE-319

## Description

The product transmits sensitive or security-critical data in cleartext in a communication channel that can be sniffed by unauthorized actors.

## Extended Description

N/A

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2022-29519: Programmable Logic Controller (PLC) sends sensitive information in plaintext, including passwords and session tokens.

**•** CVE-2022-30312: Building Controller uses a protocol that transmits authentication credentials in plaintext.

**•** CVE-2022-31204: Programmable Logic Controller (PLC) sends password in plaintext.

**•** CVE-2002-1949: Passwords transmitted in cleartext.

**•** CVE-2008-4122: Chain: Use of HTTPS cookie without "secure" flag causes it to be transmitted across unencrypted HTTP.

**•** CVE-2008-3289: Product sends password hash in cleartext in violation of intended policy.

**•** CVE-2008-4390: Remote management feature sends sensitive information including passwords in cleartext.

**•** CVE-2007-5626: Backup routine sends password in cleartext in email.

**•** CVE-2004-1852: Product transmits Blowfish encryption key in cleartext.

**•** CVE-2008-0374: Printer sends configuration information, including administrative password, in cleartext.

**•** CVE-2007-4961: Chain: cleartext transmission of the MD5 hash of password enables attacks against a server that is susceptible to replay (CWE-294).

**•** CVE-2007-4786: Product sends passwords in cleartext to a log server.

**•** CVE-2005-3140: Product sends file with cleartext passwords in e-mail message intended for diagnostic purposes.

## Related Attack Patterns (CAPEC)

* CAPEC-102
* CAPEC-117
* CAPEC-383
* CAPEC-477
* CAPEC-65

## Attack TTPs

**•** T1040: Network Sniffing (Tactics: credential-access, discovery)

**•** T1056.004: Credential API Hooking (Tactics: collection, credential-access)

## Modes of Introduction

**•** Architecture and Design: OMISSION: This weakness is caused by missing a security tactic during the architecture and design phase.

**•** Architecture and Design: For hardware, this may be introduced when design does not plan for an attacker having physical access while a legitimate user is remotely operating the device.

**•** Operation: N/A

**•** System Configuration: N/A

## Common Consequences

**•** Impact: Read Application Data, Modify Files or Directories — Notes: Anyone can read the information by gaining access to the channel being used for communication. Many communication channels can be "sniffed" (monitored) by adversaries during data transmission. For example, in networking, packets can traverse many intermediary nodes from the source to the destination, whether across the internet, an internal network, the cloud, etc. Some actors might have privileged access to a network interface or any link along the channel, such as a router, but they might not be authorized to collect the underlying data. As a result, network traffic could be sniffed by adversaries, spilling security-critical data.

**•** Impact: Read Application Data, Modify Files or Directories, Other — Notes: When full communications are recorded or logged, such as with a packet dump, an adversary could attempt to obtain the dump long after the transmission has occurred and try to "sniff" the cleartext from the recorded communications in the dump itself. Even if the information is encoded in a way that is not human-readable, certain techniques could determine which encoding is being used, then decode the information.

## Potential Mitigations

**•** Architecture and Design: Before transmitting, encrypt the data using reliable, confidentiality-protecting cryptographic protocols. (Effectiveness: N/A)

**•** Implementation: When using web applications with SSL, use SSL for the entire session from login to logout, not just for the initial login page. (Effectiveness: N/A)

**•** Implementation: When designing hardware platforms, ensure that approved encryption algorithms (such as those recommended by NIST) protect paths from security critical data to trusted user applications. (Effectiveness: N/A)

**•** 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. (Effectiveness: N/A)

**•** Operation: Configure servers to use encrypted channels for communication, which may include SSL or other secure protocols. (Effectiveness: N/A)

## Applicable Platforms

**•** None (Class: Not Language-Specific, Prevalence: Undetermined)

## Demonstrative Examples

**•** Though a connection is successfully made, the connection is unencrypted and it is possible that all sensitive data sent to or received from the server will be read by unintended actors.

**•** Multiple vendors used cleartext transmission of sensitive information in their OT products.

**•** N/A

**•** The JSON result might be:

## Notes

**•** Maintenance: The Taxonomy\_Mappings to ISA/IEC 62443 were added in CWE 4.10, but they are still under review and might change in future CWE versions. These draft mappings were performed by members of the "Mapping CWE to 62443" subgroup of the CWE-CAPEC ICS/OT Special Interest Group (SIG), and their work is incomplete as of CWE 4.10. The mappings are included to facilitate discussion and review by the broader ICS/OT community, and they are likely to change in future CWE versions.

**•** Other: Applicable communication channels are not limited to software products. Applicable channels include hardware-specific technologies such as internal hardware networks and external debug channels, supporting remote JTAG debugging. When mitigations are not applied to combat adversaries within the product's threat model, this weakness significantly lowers the difficulty of exploitation by such adversaries.