**VULNERABILITIES AND ATTACKS ON TETRA**

1. **CVE-2022-24401**

A security vulnerability or an attack scenario concerning the TETRA (Terrestrial Trunked Radio) communication system, particularly with respect to keystream reuse in encrypted traffic when utilizing the TEA (Tiny Encryption Algorithm) keystream generator. The vulnerability appears to be derived from the manner in which IVs are generated and maintained, which can be leveraged by an active actor to induce keystream reuse and decrypt encrypted traffic.

System administrators and security professionals must be cognizant of security vulnerabilities such as this one to mitigate them, as they can have a detrimental effect on the security and confidentiality of encrypted communications within systems such as TETRA.

This attack scenario serves as a reminder of the need for reliable key management and intravenous (IV) generation within encryption systems. It is essential to generate IVs in a manner that prevents the formation of predictable patterns and the reuse of IVs across multiple messages or frames. Furthermore, robust authentication and integrity verification systems can help to protect against such attacks. Security researchers and organizations often work to identify and mitigate these types of threats to enhance the security of communication systems.

FIGURE 1

Figure 1, This attack works by spoofing unauthenticated frame number update messages which function to synchronize network time, in this manner an attacker can recover keystreams corresponding to a specific moment in time t by collecting encrypted traffic of interest. An attacker can then attack any radio system with the shared keys at a later point in time to recover the key stream and decrypt the earlier collected traffic. An attacker can recover voice communication and inject data. 

A diagram of a keyboard

Description automatically generated

There are two main headers: the MAC header, which is transmitted in cleartext, and the LLC header which encodes the properties of the link layer.

The MAC header is responsible for transmitting essential information such as the recipient's SSI, the encryption of the message, and the length of the message. The fill bits flag allows for flexibility in the length of messages, indicating whether they are padded.

The LLC header, however, contains a variety of information, including the link layer type, the FCS (Frame Check Sequence) for error control, if the message requires confirmation, and if it contains a confirmation of a previous message. Additional information may be added to the LLC header depending on the type of the LLC PDU.

The frame check sequence (FCS) is an element used for error detection in the protocol. If a valid FCS exists, it is verified and removed prior to the message being passed to higher-level protocol handler functions. The presence or absence of an FCS has the equivalent effect for these higher-level functions.

The information also distinguishes between two link types, Basic Link and Advanced Link. For Advanced Link, the FCS is required, while for Basic Link it is optional. However, in the context of the information provided, the services that use the Basic Link are the primary focus, making the distinctions between the two link types less important.

In summary, this information provides an insight into the technical complexities of formatting and transmitting data over the air interface in a communication system context, highlighting the importance of headers such as MAC and LLC, and the FCS's role in ensuring data integrity.

**Recovering uplink keystream**

An attacker aims to decrypt an encrypted message (c) that was sent at a specific timestamp (t) using cryptographic keys that have not been updated. The attacker's approach involves impersonating a network entity (SwMI) and tricking a mobile station (MS) into reusing the Initialization Vector (IV) employed at the original time (t). This is achieved by sending synchronization (SYNC) and system information (SYSINFO) frames with timestamps set just before time t.

When the MS, under the influence of the attacker, subsequently transmits a message (c') at the spoofed time (t), it is encrypted using the same keystream as the original target message (m). Therefore, any knowledge gained about the contents of c' directly translates into knowledge about m. This implies that if the attacker can determine the content and characteristics of c', they can infer the content of m.

One critical aspect is the potential for high confidence in identifying parts or all of c' based on its length and its alignment within the observed communication stream. In many cases, the length and positioning of c' may be characteristic of certain types of communication, enabling the attacker to establish significant information about m.

This underscores the importance of robust security measures in communication systems to protect against impersonation attacks and the reuse of cryptographic parameters, such as IVs. It also emphasizes the significance of encryption and authentication mechanisms to thwart such adversarial attempts to gain access to confidential messages. In practice, to mitigate this type of vulnerability, implementing strong authentication, secure key management, and cryptographic protocols that prevent IV reuse is crucial for ensuring the confidentiality and integrity of communications.

**Recovering downlink keystream**

This method outlines information for recovering the downlink keystream in a communication system, despite the challenge posed by the SwMI's perception of time being beyond the attacker's control. Downlink keystream recovery is typically less straightforward than uplink recovery, but this approach demonstrates its feasibility by sending ciphertexts to the mobile station (MS) and observing its behavior in response.

The objective is to expand the knowledge of a known keystream (ks) at a specific time (t) by constructing a message (m) at the Logical Link Control (LLC) layer with the same bit length as ks plus one bit. This concept is referred to as "keystream expansion." The constructed message (m) is of type BL-DATA with FCS as its Protocol Data Unit (PDU) type, indicating the presence of the FCS checksum and requiring acknowledgment upon correct reception. The actual contents of m are irrelevant and are filled with zeroes. Additionally, m concludes with the FCS checksum computed over its contents.

To create the expanded keystream (ks'), a zero bit is appended to the known keystream (ks), resulting in ks' = ks ⧺ 0. The encrypted message (c) is then generated as c = m ⊕ ks'.

The process continues by sending spoofed synchronization (sync) and system information (sysinfo) frames, setting the MS's timestamp to slightly precede t, and transmitting message c exactly at time t. Depending on whether the FCS is correct, and, by extension, whether the newly added zero bit in ks' is the correct keystream bit, the MS will either emit an acknowledgment or silently discard the message. In either case, the attacker learns a keystream bit. This procedure is repeated continuously until the entire downlink keystream at time t is eventually learned.

This method highlights the potential vulnerabilities in communication systems and the importance of strong encryption, authentication, and integrity mechanisms to protect against keystream recovery attacks. It also emphasizes the need for robust security measures to mitigate the risks associated with the exploitation of communication protocol behaviors.

1. **CVE-2022-24402**

This is a major security issue with regard to the TEA1 Keystream Generator. It appears that the key register initialization function of the generator reduces the length of the key to only 32 bits, which significantly weakens the system's security. This is due to the fact that modern cryptographic attacks can be quickly **brute forced** with current computing power.

Therefore, it is recommended to use keys that are longer and more complex, ideally closer to the full length of 80 bits. Furthermore, key management practices such as secure key generation and storage should be adhered to reduce the risk of a key compromise.

The overall conclusion is that the implementation of a truncated key in TEA1’s keystream generator poses a risk to the system’s ability to withstand cryptographic attacks and indicates a need for a more robust key management strategy to address potential security vulnerabilities.

**Recovering full keys**

This is a fundamental security vulnerability in the TETRA encrypted system, particularly when it comes to the handling of encryption keys. This vulnerability is due to the compression of the extended key (EGK) into a 32-bit length, which not only reduces the length of the key but also presents a number of serious security risks.

This compression function assigns a reduced EGC to each channel within the network, each of which has its own EGC. The two main consequences of this vulnerability are that a reduced EGC on a particular channel allows for encryption and decoding operations only within that channel. More importantly, if an attacker can access the entire 80-bit EGC (DCK, SCK, CCCK, or MGCK), they are able to gain complete control over the network's communications, allowing for unauthorized decryption, interception, and even the forging of communications.

The complexity of the attack is further exacerbated by the fact that it is relatively simple. The compression function has specific properties that allow an attacker to generate a large number of candidate keys efficiently. The attack complexity is 2^48, which is indicative of the vulnerability's severity. To be successful in the attack, the attacker must be able to distinguish between correct key guesses and incorrect ones.

The attack approach involves recovering three different reduced ECCs from different cell or carrier frequencies. The network information transmitted to the compression function is different for each of these. The attacker then iterates over the potential pre-image of the first reduced ECC. The inverse of TB5 is then applied and the resulting key is verified to generate the other two reduced ECCs after applying TB5 again with varying network information parameters. Finally, the attacker is able to determine the entire 80-bit key.

This information highlights a fundamental vulnerability within TETRA's encryption system, wherein a decrease in the bit length of the Encryption Key (ECK) can result in a network-wide breach of security, if the attacker is able to recover the entire key. To mitigate this risk, it is necessary to reevaluate TETRA's key compression function, as well as the general key management procedures within the system, to protect the security and confidentiality of network communications. Implementing more robust encryption protocols and key management procedures are necessary to effectively mitigate this risk.

1. **CVE-2022-24404**

The lack of cryptographic integrity check in air-interface encrypted TETRA traffic poses a major and concerning security risk. Securing data is as important as its confidentiality in secure communication systems, and a cryptographic integrity check (MAC or checksum) is essential for the recipient to be able to verify that the data has not been altered during transmission. Without such a mechanism in the encryption strategy of TETRA, the data is vulnerable to potential attacks that could compromise its integrity, potentially leading to far-reaching repercussions.

Stream ciphers are a form of encryption that works by combining a keystream with plaintext to create a ciphertext. While stream ciphers provide a means of safeguarding data, they do not provide a means of verifying its accuracy after decryption. This is especially problematic when an active adversary is able to intercept and manipulate the encrypted ciphertext.

A malicious actor with the ability to intercept and alter data in transit can take advantage of this vulnerability to alter the encryption code bit by bit. By manipulating individual bits of the encryption code, they can manipulate the decoded plaintext and potentially corrupt it, misinterpret it, or even inject malicious content into it. In other words, they can alter the intended communication and cause confusion, disinformation, or even use the compromised data for malicious purposes.

Furthermore, the absence of cryptographic integrity controls not only compromises data integrity, but also data confidentiality. By manipulating the encryption code and observing the corresponding alterations in the decoded data, an attacker may gain insight into the contents or patterns of encrypted information that should remain confidential.

To effectively address this critical problem, it is essential to implement a reliable cryptographic integrity check within TETRA. This check is intended to identify any unauthorized changes or attempts to tamper with data while it is being transmitted. Generally, this can be achieved through the use of HMACs or encrypted encryption modes that combine encryption with integrity checks. These techniques can help to guarantee the integrity of data and quickly identify and reject any attempts to tamper.

Maintaining the security of communications systems is a continuous process. Regular changes and revisions of encryption algorithms, protocol specifications and key management procedures are necessary to address known weaknesses and potential threats, thus safeguarding the system in the long term.

To sum up, the lack of cryptographic integrity checks in encrypted traffic within TETRA airspace poses a significant risk to data integrity, as it allows active adversaries to tamper with data. Effective integrity checks are essential to prevent unauthorized modifications and protect the integrity and confidentiality of communications and should be implemented as a best practice to enhance the security of all communication systems, particularly in cases where the integrity of data is of utmost importance.

MITIGATION - The recommendation to Implement End-to-End Encryption (E2E) is an effective means of mitigating the security and privacy risks associated with the transmission of data in communication systems. E2E encrypts data on the sender's end and only decrypts it on the recipient's end. This means that, even if an attacker were to intercept the data while it is in transit, they would not be able to decrypt its contents without the corresponding decryption key. As a result, the encryption of the communication would protect its confidentiality and integrity, making it difficult for adversaries to manipulate or intercept the data.

In addition to E2E, Operational Security (OPSEC) compensating controls (OPSEC) are beneficial for reinforcing overall security. By conducting a comprehensive risk assessment, organizations are able to tailor their OPSEC practices to the specific threats and vulnerabilities they are facing. This allows for the identification of weaknesses in the security posture and the development of tailored measures to address those vulnerabilities.

It is essential to bear in mind that, while encryption is highly effective in safeguarding data in transit, it may not be sufficient to address all security issues. OPSEC measures can complement encryption by addressing additional aspects of security, including physical access control, staff training, and security incident response.

In conclusion, the recommended mitigation is to use encryption to protect against data interception and manipulation, however, it should be incorporated into a comprehensive security strategy, which includes operational security practices that are tailored to risks and vulnerabilities. Such a layered approach to security enables organizations to address a wider range of security issues and maintain a strong defense against threats.