SECURITY ADVISORY

Remote Buffer Overflow in StrongSwan + TKM

Affected versions: **<= 5.9.11**

CVE-2023-41913

November 20, 2023

Florian Picca

Stackered

1 Vulnerability summary

1.1 Affected product

StrongSwan is an open-source, full-featured IPsec-based VPN (Virtual Private Network) solution. It's commonly used for creating secure site-to-site connections or for remote access to a corporate network.

The Trusted Key Manager (TKM) is a minimal Trusted Computing Base which implements security-critical functions of the IKEv2 protocol. The TKM works in conjunction with the strongSwan IKEv2 daemon `charon-tkm` to provide key management services for IPsec.

1.2 Description and impact

The TKM-backed version of the charon IKE daemon (`charon-tkm`) doesn't check the length of received Diffie-Hellman public values before copying them to a fixed-size buffer on the stack, causing a buffer overflow that could potentially be exploited for remote code execution by sending a specially crafted and unauthenticated IKE_SA_INIT message.

1.2.1 Affected versions and configuration

All strongSwan versions since 5.3.0, up to and including 5.9.11 are affected.

Setups that don't use `charon-tkm` as IKE daemon are not vulnerable. The `charon-tkm` version that supports multiple key exchanges (`tkm-multi-ke` branch on GitHub) is not vulnerable either.

1.3 Credits

All credit goes to Florian PICCA who found this vulnerability during his spare time.

1.4 Timeline

08/07/2023	Vulnerability reported to security@strongswan.org.
11/07/2023	Confirmation of the vulnerability and patch sent by Tobias Brunner.
20/11/2023	Public release.

2 Technical details

StrongSwan utilizes a plugin-based architecture, which means specific software modules are dedicated to performing various tasks within the VPN framework. In this setup, individual plugins are responsible for implementing cryptographic functions, key exchange protocols, and security measures.

During the IKE_SA_INIT phase, the Diffie-Hellman key exchange occurs, where both parties (the initiator and the responder) exchange their public keys. This exchange allows them to agree upon a shared secret that will be used to derive the keys necessary for secure communication.

Since StrongSwan version 5.3.0, plugins implementing Diffie-Hellman key exchanges are responsible for verifying the correctness of the received public values in their `set_public_key` function using the helper function `key_exchange_verify_pubkey`.

In the case of a TKM-backed version of the charon IKE daemon (`charon-tkm`), which acts as proxy for Diffie-Hellman operation between the IKE daemon and the Trusted Key Manager, this helper function is not called resulting in an unchecked `memcpy`.

```
METHOD(key_exchange_t, set_public_key, bool,
    private_tkm_diffie_hellman_t *this, chunk_t value)
{
    dh_pubvalue_type othervalue;
    othervalue.size = value.len;
    memcpy(&othervalue.data, value.ptr, value.len); // Insecure memcpy

    return ike_dh_generate_key(this->context_id, othervalue) == TKM_OK;
}
```

Code 1: src/charon-tkm/src/tkm/tkm_diffie_hellman.c

The `dh_pubvalue_type` type is not defined in StrongSwan's source code but in that of the TKM-RPC library.

```
typedef struct dh_pubvalue_type dh_pubvalue_type;

struct dh_pubvalue_type {
    uint32_t size;
    byte_t data[512];  // Static buffer on the stack
};
```

Code 2: tkm-rpc/specs/c/tkm/types.h

Because the length of the peer's Diffie-Hellman public key is only limited by the maximum length for accepted IKE messages, which defaults to 10'000 bytes, a buffer overflow can be triggered.

```
/**
 * Maximum packet size we handle by default
 */
#define PACKET_MAX_DEFAULT 10000
```

Code 3: src/libstrongswan/networking/packet.h

Additionaly, as this key exchange happens during the IKE_SA_INIT packet, the peer is not yet authenticated, making this vulnerability triggerable without needing a legitimate client certificate.

A successfull exploitation could result in remote code execution in the context of `charon-tkm`. The vulnerability can also be abused to repeatedly crash the daemon, causing a denial of service.