corrata

Traffic Analysis

January 2025

Summary

Information about individual users’ search and viewing history on the Xiaohongshu social media platform is being transmitted without encryption compromising their privacy and security.

Incident Type: Insecure TCP Traffic

Our analysts have investigated system reports in relation to HTTP TCP traffic between the Xiaohongshu Android app and its related servers on our customers' devices.

Overview

Upon registering with the Xiaohongshu application and viewing videos, the application initiates HTTP GET requests to a content delivery network (CDN) host such as `sns-na-i9.xhscdn.com`.

This domain, among potentially other CDN hosts, appears to serve image resources in formats like WEBP, JPG, PNG, and possibly a proprietary format referred to as REIF in the HTTP request and response header.

A critical observation which prompted us to investigate this traffic is that the network traffic for these requests is not encapsulated in neither TLS nor mmTLS, nor a proprietary encryption or encoding format. The traffic is sent in plaintext.

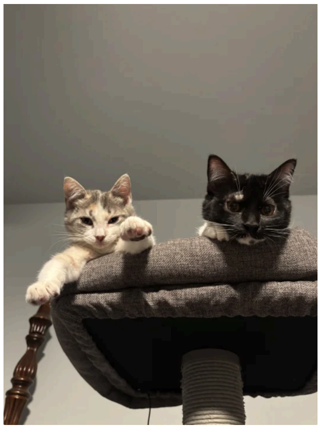
This allows adversaries observing network traffic to view the thumbnails of the videos a user is viewing and correlate this information to their activity on the app. Furthermore, plaintext HTTP does not guarantee authenticity and as such, traffic may be tampered or spoofed.

Replication of Findings

A search for “pets” yielded HTTP GET requests as below:

```
GET /REDACTED?imageView2/2/w/540/format/webp|imageMogr2/strip&redImage/frame/0
HTTP/1.1
Referer: https://app.xhs.cn/
User-Agent: REDACTED
Host: sns-na-i9.xhscdn.com
Connection: Keep-Alive
Accept-Encoding: gzip
```

Extracting TCP response traffic segments by the webp header *52494646 080D0200* yielded results such as:

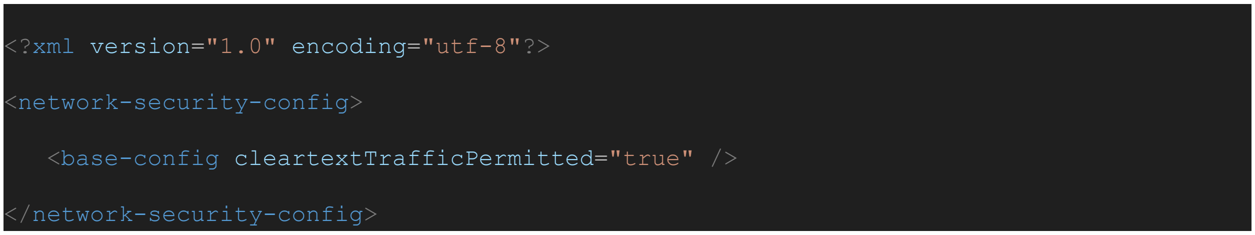


Searching for videos varies in the traffic count but most searches yield upwards of 40 plaintext HTTP frames. Registering for the application yielded upwards of 16,000 frames.

Recommended Action

Should this traffic architecture be unintentional, it is recommended to encapsulate all network traffic on applications in TLS (defined in RFC 5246).

Furthermore, see extract below from xml definitions in your application version 8.69.0, SHA256 hash e921c5908ec3d368b5c4cb7eab119f467bb97202569ad7dc19fb2a51a3f22cf8.



```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
   <base-config cleartextTrafficPermitted="true" />
</network-security-config>
```

It is not recommended to permit cleartext traffic to all domains in your base configuration. From the Android developer's documentation (see here)

"Allowing cleartext network communications in an Android app means that anyone monitoring network traffic can see and manipulate the data that is being transmitted. This is a vulnerability if the transmitted data includes sensitive information such as passwords, credit card numbers, or other personal information.

Regardless of if you are sending sensitive information or not, using cleartext can still be a vulnerability as cleartext traffic can also be manipulated through network attacks such as ARP or DNS poisoning, thus potentially enabling attackers to influence the behavior of an app."

Should you have any questions about our findings or wish to follow-up, please feel free to get in contact.