Viking Horde: a New Type of Malware on Google Play

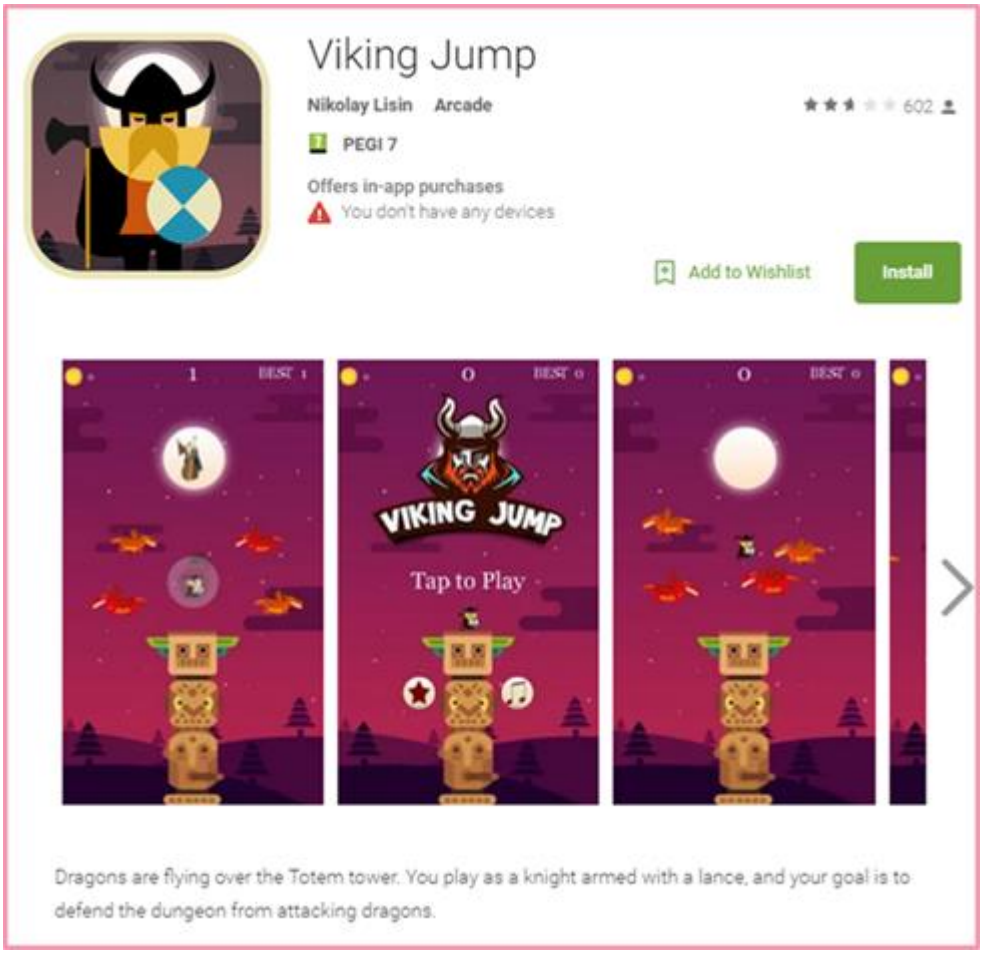
By: **Andrey Polkovnichenko and Oren Koriat**

The Check Point research team uncovered a new malware campaign on Google Play it calls Viking Horde. Viking Horde conducts ad fraud, but can also be used for other attack purposes such as DDoS attacks, spam messages, and more. At least five instances of Viking Horde managed to bypass Google Play malware scans so far.

Check Point notified Google about the malware on May 5, 2016.

On all devices -- rooted or not -- Viking Horde creates a botnet that uses proxied IP addresses to disguise ad clicks, generating revenue for the attacker. A botnet is a group of devices controlled by hackers without the knowledge of their owners. The bots are used for various reasons based upon the distributed computing capabilities of all the devices. The larger the botnet, the greater its capabilities.

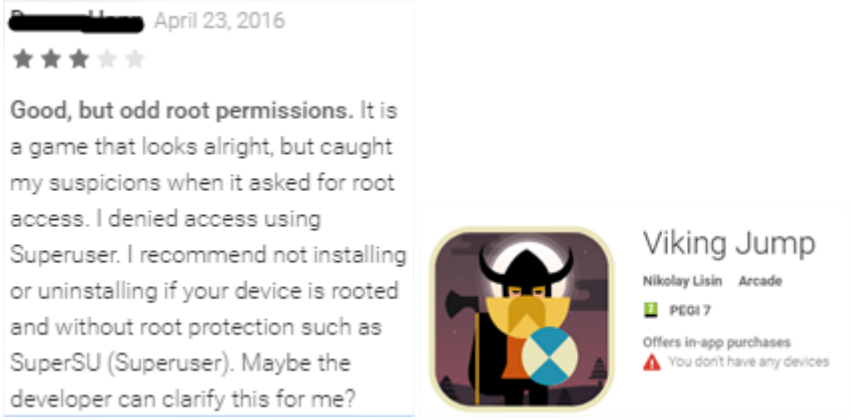
On rooted devices, Viking Horde delivers additional malware payloads that can execute any code remotely, potentially compromising the security of data on the device. It also takes advantage of root access privileges to make itself difficult or even impossible to remove manually.



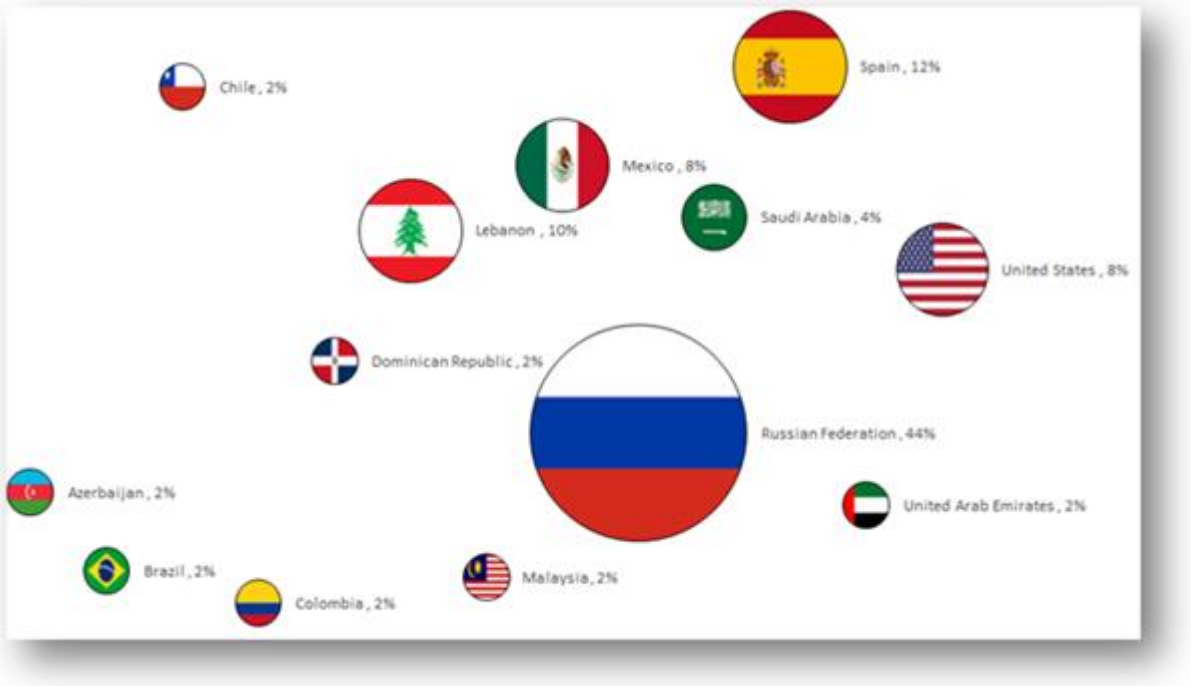
Meet the Horde

The most widely-downloaded instance of Viking Horde is the app Viking Jump which was uploaded to Google Play on April 15 and has between 50,000-100,000 downloads. In some local markets, Viking Jump is a Google Play top free app. The oldest instance is Wi-Fi Plus which was uploaded to Google Play on March 29. Other instances include the apps Memory Booster, Parrot Copter, and Simple 2048.

All Viking Horde-infected apps have a relatively low reputation which the research team speculates may be because users have noticed the app's odd behavior, such as asking for root permissions.



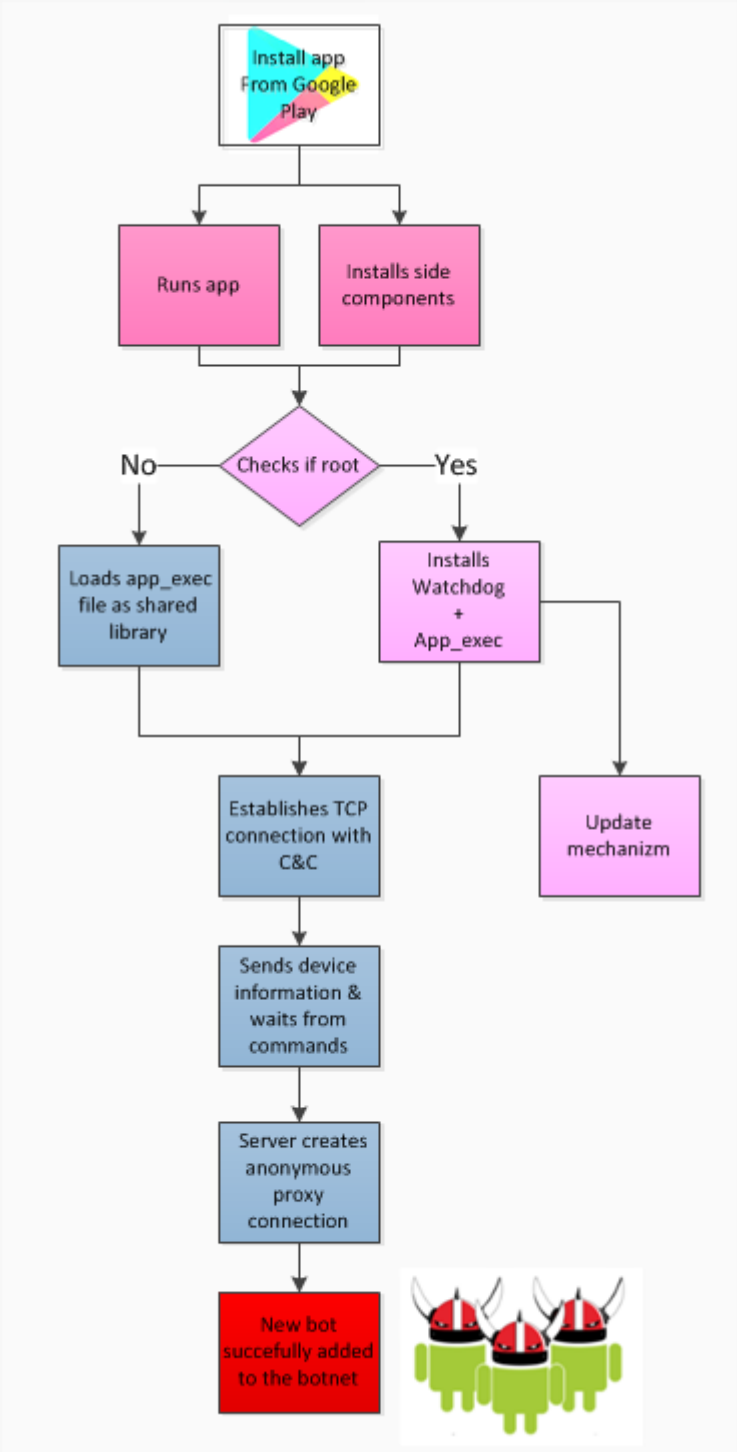
The botnet created by the attackers spread worldwide to users from various targeted countries. The Check Point research team collected data on the distribution of victims from one of the many Command & Control servers (C&C's) used by attackers, which is illustrated below:



Source: Check Point Mobile Threat Research Team, May 6, 2016

Viking Horde Operation

From its research of Viking Horde's code and the C&C servers used in the attack, the research team can illustrate the malware process flow.



1. The malware is first installed from Google Play. While the app initiates the game, it installs several components, outside of the application's directory. The components are randomly named with pseudo-system words from a preset list, such as core.bin, clib.so, android.bin and update.bin. They are installed on the SD card if the device is not rooted, and to root/data if it is. One of these files is used to exchange information between the malware's components. A second file contains the list of the generated names of the components, to make them available to all components.

2. The malware then checks whether the device is rooted:
    - If the device is rooted, the malware initiates two additional components:
        - app_exec. Implements communication protocol with the server.

    - app_exec_watch_dog Binary, implements update and persistency mechanism. Watchdog monitors app_exec process and restarts it if needed.
- If the device is not rooted, the malware loads app_exec file as a shared library and calls its functions by JNI - Java Native Interface, which allows java code run native binaries.

In both scenarios, once app_exec application is installed, it establishes a TCP connection with the C&C server and starts the communication. The communication consists of the following commands:

- Ping. Every 10 seconds application sends 5 bytes to server. Server responds with the same 5 bytes.
- Update of device information: Sends to server charge battery, type of connection and phone number.

3. The next step is to accomplish the main malicious functionality by creating an anonymous proxy connection. The C&C sends a “create_proxy” command with two IP addresses and ports as parameters. These IP addresses are used to open two sockets one for a remote server (which is a client of the botnet exploiting the anonymous proxy) and the other for the remote target. Then it reads the data received from the first socket and channels it to the target host. Using this technique, the malware developer (or someone using this botnet as “malware as a service”) can hide his IP behind the infected device’s IP.

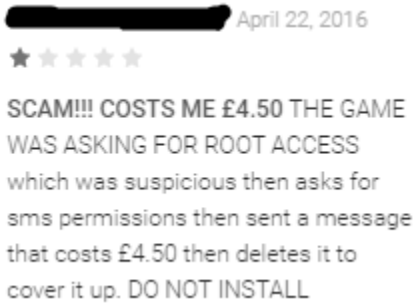
Botnet activity

It is important to understand that even if the device is not rooted, Viking Horde turns the into a proxy capable of sending and receiving information per the attacker’s commands. Below is an example of an infected device as seen from an attacker’s C&C.
The remoteIP is the proxy’s IP and the socksIP is the C&C server’s IP. The C&C contains some information about the device including its OS version, battery status, and GPS coordinates. In this case, the device is located in the US on T-Mobile.


```
"remoteIP": "172.58.17.116",
"socksIP": "5.9.240.84",
"socksPort": 53890,
"deviceInfo": {
    "versionProtocol": 3,
    "installType": 0,
    "deviceId": "a6c457eb174b1f22",
    "timeZone": -6,
    "versionOS": 22,
    "platform": 1,
    "versionSoftware": 5,
    "networkType": 0,
    "batteryStatus": 0,
    "phoneNumber": "16239862043",
    "geo": {
        "country": {
            "name": "United States",
            "code": "US"
        },
        "isp": "T-Mobile USA"
    }
},
"onlineTime": 4715168
```


The botnet is controlled by many C&C servers, each managing a few hundred devices. The malware’s primary objective is to hijack a device and then use it to simulate clicks on advertisements in websites to accumulate profit. The malware needs this proxy to bypass

ad-nets' anti-fraud mechanisms by using distributed IPs. (Ce PDF est diffusé par zataz.com)
Some user reviews of the app also claim it sends premium SMS messages, as seen in the screen capture below. This botnet could be used for various malicious purposes, such as DDoS attacks, spamming and delivering malware.

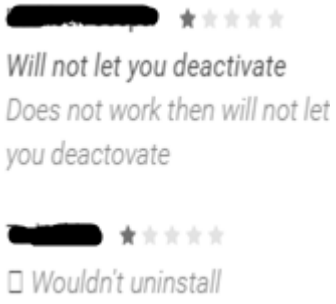


Vikings are a persistent Horde

The malware uses several techniques to remain on the device. First, Viking Horde installs several components with system-related names, so that they are hard to locate and uninstall.
If the device is rooted, two more mechanisms are set in place:

- The app_exec component monitors the main application's existence. If the user uninstalled the main application, app_exec decrypts a component called com.android.security and silently installs it. This component will be hidden, and run after boot. This component is actually a copy of itself, and has the same capabilities.
- The watchdog component installs the app_exec component updates. If app_exec is removed, the watchdog will reinstall it from the update folder.

Apparently, some users even noticed this activity:



Bonus component for rooted devices

Perhaps the most dangerous functionality is the update mechanism. The update mechanism is split between app_exec and watchdog components. app_exec downloads the new binary from the server and stores it to /data directory with app_exec_update name.

Watchdog periodically checks if an update file exists and replaces app_exec with this file. This means that upon the server's command, Viking Horde downloads a new binary. The watchdog component will replace the application with it. This allows downloading and executing any remote code on device.

Appendix 1: app package names

com.Jump.vikingJump
com.esoft.wifiplus
com.fa.simple2048
com.android.wifiman
Com.g.o.speed.memboost
Com.f.a.android.flyingcopters

Appendix 2: list of C&C servers:

www[.]adautoexchange.com
www[.]adexchng.com
www[.]adexchnge.com
www[.]adexchangetech.com

Appendix 3: Infected binaries SHA256:

85e6d5b3569e5b22a16245215a2f31df1ea3a1eb4d53b4c286a6ad2a46517b0c
254c1f16c8aa4c4c033e925b629d9a74ccb76ebf76204df7807b84a593f38dc0
ebfef80c85264250b0e413f04d2fbf9e66f0e6fd6b955e281dba70d536139619
10d9fdbe9ae31a290575263db76a56a601301f2c2089ac9d2581c9289a24998a
a13abb024863dc770f7e3e5710435899d221400a1b405a8dd9fd12f62c4971de
1dd08afbf8a9e5f101f7ea4550602c40d1050517abfff11aaeb9a90e1b2caea1
e284a7329066e171c88c98be9118b2dce4e121b98aa418ae6232eaf5fd3ad521