Hunting ELFs:
An investigation into Android malware detection

Authors
Thomas S Atkinson, MSc (Royal Holloway, 2016)
Lorenzo Cavallaro, ISG, Royal Holloway

Abstract

In the depths of Android mobile applications all over the world, malicious ELFs are lying dormant and hidden, awaiting activation by the malware that controls them. But fear not! Though they may lie in secret, for those who would hunt them these ELFs leave a trail that can be followed. This is the story of the hunt and how these ELFs came to reveal themselves.[a]

[a]This article is published online by Computer Weekly as part of the 2017 Royal Holloway information security thesis series `http://www.computerweekly.com/ehandbook/Hunting-ELFs-An-investigation-into-Android-malware-detection`. It is based on an MSc dissertation written as part of the MSc in Information Security at the ISG, Royal Holloway, University of London. The full MSc thesis is published on the ISG's website at `https://www.royalholloway.ac.uk/isg/research/technicalreports/technicalreports.aspx`.

When we talk about ELFs we are of course referring to the Linux executable files: Executable and Linkable Format (ELF). These are equivalent to .exe files on Windows. In the context of Android and Android malware, these ELFs are just as tricky as their pointy-eared counterparts. As most people are aware, Android apps are primarily written in the JAVA programming language with the ability to invoke native code. All the code and files used by the app is bundled into a .APK file. This APK (Android application package) is the file your device downloads from Google Play or third party stores when you download and install an app. As well as using JAVA to run code, the app can also call ELF files. All of this is still stored within the app's APK file and is invisible to the end user. There can be many legitimate reasons to use ELFs as part of an app, such as code re-use or hiding proprietary code. As it happens though, the use of ELF files alone is a pretty good indication that the app may be up to something and may not be quite what it purports to be.

The hunt begins

In the beginning of the hunt, we started by looking at known malicious apps with ELF files in them. We knew from previous research that ELF files were a prominent feature amongst some Android malware and that any app with an ELF should definitely be investigated. We started by dissecting these ELFs (don't worry - no ELFs were harmed in the writing of this paper) to see what they were up to and what common features could be gleaned from their innards. Just like Windows' .exe files, ELFs are made up of sections that have special flags set to tell the computer what to do with each section. During our autopsy on these captured ELFs, some interesting features began to reveal themselves to us. We found that some seemingly benign ELFs had some weird sections with strange flags set on them.

Clue #1

There are ELFs.

Suspicious flags

Previous research into hidden malware in ELFs and .exe files on desktop computers had found that some malware hid its malicious code in a section of the executable as packed data. By this we mean that the data was either encrypted or compressed (like a .zip file for example). These sections then had the Execute (X), Allocate (A) and Read (R) flags set. When the operating system needs to load an ELF, it needs to know what sections are loadable and thus need space allocated for them (A), and

whether the section is executable (X) and/or readable (R). This is everything required to unpack and run the malware hidden in the section. These techniques are traditionally how malware hid itself within a seemingly benign file. What we found was a far more devious and sinister technique hitherto unknown. We found sections within seemingly benign ELFs that contained full path names to other files within the APK and nothing else in the section. The flags on these sections were very strange and not usually seen together. Allocate (A) - used to allocate new memory, Merge (M) - used to merge this section with other sections in the ELF, and Strings (S) - this basically just means that this section is a special kind of text data only. So these three flags, AMS, were the first half of the trail to finding the malicious ELFs.

Clue #2

The ELFs have full paths names to other files with strange flags.

Trails to dubious ends

As previously mentioned, these ELF files we were dissecting did not immediately seem malicious in themselves and contained no dangerous calls to system functions or other such trickery. The only clue that something was amiss were the three flags AMS and the full paths to other files in the APK. The next stage was to examine the files named within these suspect ELFs.

Subtlety is an art form lost on modern day malware writers, it would seem. Quite a few of the files whose paths were contained in the seemingly benign ELFs had obviously dodgy names. Some were called “rat” which is an acronym for Remote Access Tool. RATs are used by hackers and malware authors as a backdoor to remotely control a device. Another name we saw was “rageinthecage”, which is the name of a well known Android exploit used to break out of the sandbox that all apps run in. So it was clear that these files were very likely malware.

The next stage in the hunt was to examine these files in greater depth and see what was inside them. When we opened them up and peered inside we found . . . random data! This may look like a dead end but it was in fact the last stage of the hunt that confirmed we had found hidden malicious ELFs. Most files are not random and although an executable may look random when you opened it up, it is in fact made up of a series of set instructions that are organised in a somewhat predictable manner. As alluded to earlier, only encrypted or compressed files have very random data. There is a branch of mathematics called entropy that allows us to assign a value to randomness. A truly organised and repetitive file would have an entropy of zero and a totally random file would have an entropy of 8 (if we are looking at bits per byte of entropy). In reality neither of these values are often seen as most things are not completely organised or completely random. In reality, a value of 1-3 is very organised and a value of 7 or higher is very random. The entropy value of the files we scanned were well over 7.5. This confirms the suspicion that these files were packed (either compressed or encrypted).

Clue #3

The files have very high entropy.

A reasonable person might well ask whether this method is just a way for legitimate software to hide itself from being read by competitors. To investigate this we scanned 3,029 ELF files from a well used Linux computer and none of them contained calls to packed binaries. Although this may not provide statistically-supported evidence, it is a reasonable approximation at this early stage of the research.

So now we have the full trail to the malicious ELFs. The first sign is the presence of an ELF file. The next sign is a section in the ELF containing full path names to other files in the APK with the flags Allocate (A), Merge (M) and Strings (S) set. The final piece is that these files named have an entropy higher than 7 and are therefore packed files.

The hunt continues

Now the hunt was on. This new-found information about malicious ELFs can be fed into Android malware detection systems to increase the accuracy of detection. Performing large-scale evaluations is part of our ongoing future work which will support our findings with statistically-supported evidence

on the nature of ELFs in malicious Android apps.

Biographies

Thomas S Atkinson spent the formative years of his career trying not to blind himself and others with lasers whilst helping to design the backbone of the modern world. After four years in telecoms research and development, he left to pursue his passion for computer security and completed a Masters in Information Security at Royal Holloway for which he received a distinction for his efforts. Thomas went on to become a Security Consultant for NCC Group where he currently advises top FTSE 250 companies on their security.

Lorenzo "Gigi Sullivan" Cavallaro was raised in a fantastic epoch where information and knowledge was meant for those who were just curious enough. He grew up on pizza, spaghetti, Phrack (do "smashing the stack for fun and profit" and "IP spoofing demystified" ring a bell to you?), and W. Richard Stevens' TCP/IP illustrated masterpieces. Underground and academic research interests followed shortly thereafter and he has never stopped wondering and having fun ever since. Lorenzo is now a Reader (Associate Professor) of Information Security in the Information Security Group (ISG) at Royal Holloway, University of London. Lorenzo's research focuses largely on systems security. To this end, he has founded and is leading the recently-established Systems Security Research Lab (S2Lab) within the ISG, which focuses on devising novel techniques built around program analysis and machine learning to protect systems from a broad range of threats, including those perpetrated by malicious software. In particular, Lorenzo's lab aims ultimately at building practical tools and providing security services to the community at large. He is Principal Investigator and co-Investigator on a number of UK EPSRC- and EU-funded research projects, sits in technical program committee of well-established information security academic conferences and workshops, and has published in well-known venues. Lorenzo's Coursera MOOC on "Malicious Software and its Underground Economy: Two Sides to Every Story" attracted more than 100,000 students since its pilot in 2013, which makes him shamelessly bragging on his pizza, spaghetti, and Phrack heritage furthermore.