SSN Project Proposal

Stamatis Maritsas
Kenneth van Rijsbergen
Yadvir Singh

Abstract

On Android it may be possible to obtain the SSL session keys by scanning and parsing the process memory of a running application. We would like to investigate whether it is possible to recover the keys to decode captured network traffic of SSL sessions.

1 Research question

> *How can we obtain the SSL sessions keys from the process memory of an running Android application?*

Our research is going to include the following steps:

1. Investigate the architecture of Android processes, related work, and methods of analysing the application memory.
2. The actual analysis of memory dumps of running process and parsing the data to find keys.
3. Extract the SSL keys and try to decode captured traffic.
4. Testing different applications to examine similarities in key storage.

2 Related work

The article by Gursev Singh Kalra, titled "Extracting RSAPrivate-CrtKey and Certificates from an Android Process", describes how to dump X.509 certificates and construct a RSA private key (RSAPrivate-CrtKey) from the Android application memory using Eclipse Memory Analyzer Tool (MAT) and Java code. This paper gave us the indication that their are possibilities to extract the keys from a running process. [1]

3 Planning

Week	Work
1	Start looking at how Android processes internally work and how SSL keys are stored. Also, look at memory analysis and tools that can parse and analyse memory dumps. Furthermore do research on the SSL protocol and its implementation in android.
2	**Phase 1 of 2nd week:** Locate the cryptographic material using the chosen memory analysis tool. **Phase 2 of 2nd week:** Try to dump any possible certificates and extract the SSL keys. Repeat the process on multiple applications.
3	Finalize memory analysis and dump. Set-up test environment to capture encrypted traffic and try decrypting the traffic using the captured keys. Write down initial conclusions and repeat experiment on different applications.
4	Write the report and finalize conclusions. Start preparation of the presentation + extra time for any delays.

4 Tools and Equipment

We intend to use an Android smart phone for the memory analysis in combination with a tool to dump and inspect the memory (i.e. Eclipse Memory Analyzer). When successfully extracted a key we will use a separate router to capture and later decode the traffic.

5 Ethical issues

We will use a test phone that is owned by one of the team members. This will ensure no personal information is published without consent. We will start by trying to extract SSL keys from a simple application that does not involve personal information. If the extraction of SSL keys is successful we will try other applications that deal with personal information.

If key extraction is possible from another app on the same device than this is a serious vulnerability that should be reported via responsible disclosure. However, it is likely that physical access or administrator privileges are required for an exploit.

Although there is a risk that the results of this research might be used for malicious purposes, we feel that doing this research will contribute to the security awareness of android processes.

We also have to be sure that the software that we are going to use in order to analyse the memory of an application and the heap are not violating the terms and conditions of the application.

References

[1] Gursev Singh Kalra. Extracting rsaprivatecrtkey and certificates from an android process. `http://blog.opensecurityresearch.com/2013/10/extracting-rsaprivatecrtkey-and.html`, 2013.