Application of attack potential to FIDO L1+ Authenticator

FIDO Alliance Final Requirements Document 02 November 2021

This version:
https://fidoalliance.org/specs/fido-security-requirements/FIDO-L1+-Application-of-Attack-Potential-v1.0-fd-20211102.html
Editor:
Beatrice Peirani, Thales

Abstract

This document describes a methodology for evaluation of a software FIDO product protected with software security tools, by calculating an attack potential.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. The most recent version of this document can be found on the FIDO Alliance Website at https://fidoalliance.org.

This document was published by the FIDO Alliance Security and Privacy Working Group as a Final Requirements Document. If you wish to make comments regarding this document, please Contact Us. All comments are welcome.

Table of Contents

1. Notation

1.1 Version

This document version (DV) is DV 1.0.0.

2. Introduction

This document describes a methodology for evaluation of a software FIDO product protected with software security tools, by calculating an attack potential. To that aim, this document largely inspires from Annex B of [ISOIEC-18045], [AttackPotentialSmartcards] and Annex A of [TEE-PP].

Attack potential is a definition of ability for an attacker to perform a set of attacks. Attack potential is function of expertise, resources, access to the product and motivation of attacker to perform an attack on a given product. This document uses some factors, selected in the list of possible factors identified in [ISOIEC-18045], and completed by some adequate in a software environment.

3. Scope

This document provides a method to calculate the attack potential required by an attacker to succeed in illegally use a software L1+ FIDO product as defined in FIDO Security Reference. As such, it provides guidance for evaluation laboratories on the attack methods that have to be considered in a FIDO L1+ evaluation. Additionally by describing the factors to be considered (this document) and by detailing examples of rating (in [[!FIDO-L1+-AM]]), this document allows harmonization of conducted evaluation between laboratories.

It shall be particularly noticed that the listed attack methods are a minimum set of methods to be considered, based on commonly agreed state-of-the-art (see Bibliography). However, additional attacks may be identified by FIDO laboratories during an evaluation. Finally, not all attack methods are systematically applicable to all software FIDO products.

This document introduces attack paths which are most likely composed – in the context of software products – of several steps (namely partial attack methods or shortly partial attacks) involving various technologies and expertise. The method to calculate the attack potential specified in this document is intended to be used for **complete** attacks only (namely full attacks), i.e., taking into account all the (partial) attack steps required in the realization of the attack.

4. Glossary and Definitions

4.1 Glossary

DAST	Dynamic Application Security Testing
DBI	Dynamic Binary Instrumentation (framework)
DCA	Differential Computation Analysis
DFA	Differential Fault Analysis
TOE	Target of Evaluation
WBC	White Box Cryptography

4.2 Definitions

Dynamic Binary Instrumentation framework (DBI)

A tool to analyze into detail the behavior of a target program. Initially designed for performance or memory leak detection, this kind of feature also is of great interest for side-channel analysis (through data collection) or fault injection techniques (through dynamic data or control flow modification). NOTE: Frida (https://www.frida.re/) and Triton (https://triton.quarkslab.com/) are DBIs that are security-oriented, while Pin (https://software.intel.com/en-us/articles/pin-a-dynamic-binary-instrumentation-tool), Valgrind (http://valgrind.org/docs/valgrind2007.pdf) or DynamoRio (https://www.dynamorio.org/) have plugins meant for security analysis.

Partial Attack

A partial attack is an attack that targets specifically a feature or a mechanism of the software product. This partial attack may reveal a vulnerability of the product but by definition partial attack is not sufficient to demonstrate that the vulnerability can be exploited in a practical attack scenario (for instance with all other security mechanisms activated).

Full Attack

By opposition to a partial attack, a full attack demonstrates that vulnerabilities can be exploited in a practical attack scenario, i.e., an asset of the FIDO product can be successfully attacked. The demonstration is most likely to include the combination of several partial attacks so that several vulnerabilities are combined to defeat the security mechanisms implemented.

5. Calculating attack potential

5.1 Determining attack potential

By default in Common Criteria, determining the attack potential during a security certification shall be based on the rating table specified in Annex B of [ISOIEC-18045]. There are few internationally recognized exceptions to this generic method that are only relevant to particular technical domains namely smartcards [AttackPotentialSmartcards], hardware devices with security boxes and Trusted Execution Environment [TEE-PP].

In this document, we considered possible improvements to Annex B of [ISOIEC-18045] to improve adequacy to the specificities of software products, reusing, when relevant, principles developed in [AttackPotentialSmartcards] or Annex A of [TEE-PP].

First, similarly to [AttackPotentialSmartcards] and Annex A of [TEE-PP], the calculation of the attack potential for software FIDO products (as specified below) makes the distinction between “Identification” and “Exploitation”. Identification phase corresponds to the first creation of the attack (generally requiring most of the resource and skills of the attacker with a given attack potential), while exploitation phase corresponds to the use of the identified/developed tools/techniques to perform the attack on the field (potentially by an attacker with a lower attack potential).

Split of identification and exploitation is also relevant for software products: identification often requires a strong expertise (e.g., reverse engineering hackers), while exploitation in a connected world may be available to anyone (e.g. script kiddies) thanks to Internet published exploit tools and scripts. In the calculation specified in this document, the two attack potential for identification and exploitation phases are therefore combined to obtain the final rating.

Attack methods calculation is mapped onto relevant factors as specified in Annex B of [ISOIEC-18045]:

elapsed time,
- expertise,
- knowledge of TOE,
- windows of opportunity and,
- equipment

The document additionally introduces a new factor “Replicability” specified in below

Finally, the attack potential of the attacker of the software FIDO product is characterized as Basic, Moderate or High (as defined in Calculation of Attack Potential). FIDO L1+ products must resist to an attacker with minimum of Moderate potential.

5.2 Factors to be considered

The following factors are specified in Annex B of [ISOIEC-18045], except the “Replicability” factor.

5.2.1 Elapsed time

The time spent by an attacker to identify, prepare, develop and exploit the attack.

In practice for software FIDO products, an evaluator is unlikely to spend more than 45 days attacking the TOE.

The amount of time is as follows:

- Less than one hour
- Between one hour and one day
- Between one day and one week
- Between one week and one month
- More than one month

5.2.2 Expertise

The technical expertise required for an attacker. He can be:

- Layman: no particular expertise (“script kiddies”)
- Proficient: familiar with security behavior, classical attacks
- Expert: familiar with one of the following “software” skills:
  - implemented algorithms and protocols (or cryptography),
  - principles and concepts of security,
  - techniques and tools for the definition of new attacks including WBC (and well-known attacks like side-channel DCA or fault DFA),
  - and reverse engineering, especially in a mobile execution environment.
- Multiple Expert: familiar with several software skills (these skills must be strictly different, e.g. reverse engineering and (WB) cryptography…)

5.2.3 Knowledge of TOE

The knowledge of the design and operation of the target of evaluation. It can be:

- Public: access to public information only, e.g. available on the Internet, App Store
- Restricted: access to information given by the vendor, with restricted diffusion (e.g. covered by NDA), e.g. functional specification
- Sensitive: access to information obtained by social or reverse engineering of developers, e.g. high level design, or reverse engineering of the development kits libraries or tools that are restrictedly shared (e.g. covered by NDA)
- Critical: access to information restricted to few individuals and likely cannot be obtained through social engineering or attacks on development kits and thus require an enormous effort to be retrieved within developer information system (including with insider support).

5.2.4 Windows of opportunity

Windows of opportunity factor as specified in Annex B of [ISOIEC-18045] refers to the access opportunity to the target product that is required for the attack. In generic cases access opportunity may be estimated in access duration or quantity of product samples. However, access duration is already considered in “Elapsed Time” factor and quantity of samples is not relevant for software products. This in the context of this document access opportunity shall be estimated considering the difficulty required in accessing one or several valid functional software instances without the attack being noticed (e.g. by a FIDO product owner or administrator):

- Unlimited access: only one instance is enough or no enrolment required or unlimited enrolments are possible
- Easy access: one or few instances may be required or simple enrolment process
- Moderate access: several instances are required or enrolment is mandatory to proceed with sensitive operations
- Difficult access: significant number of instances or mandatory enrolment process completed with out of band verifications to proceed with sensitive operations
- None: require number of instances or enrolled account that cannot be obtained by an attacker.

5.2.5 Equipment

- Standard: all software tools are available on Internet or at reasonable cost
- Specialized: all software tools are either costly or need some customized development
- Bespoke: all software tools are highly sophisticated, costly, and developed for a targeted application

The following table gives some examples of tools at standard and specialized levels.

Tool	Equipment
Tools for code protection	
Smartphone emulator	Standard

Disassembler and/or decompiler (ex: IDA Pro)	Standard
Debuggers or Dynamic Instrumentation tools	Standard
Leaks or public exploits	Standard
DAST tools (ex: HCL AppScan)	Specialized
Services done by expert using its own set of customized tools	Specialized
Tools for cryptographic protections	
Side Channel Tools (ex: Side Channel Marvels)	Standard
Advanced Side Channel Tools (ex: eShard esDynamic)	Specialized
Services done by expert using its own set of customized tools	Specialized

5.2.6 Replicability

Replicability is a new factor (compared to referenced state of the art).

In the current application of the Common Criteria methodology for calculating an attack potential, the calculation estimates (1) the attacker's effort to identify the presence of a vulnerability and (2) the effort to reproduce (exploit) it on similar samples. It is particularly true in several industries where produced/manufactured products in a "family" are all identical (e.g., smart cards).

In the context of software products installed on mobile devices, each compiled/delivered product may have specificities since the mobile market is heterogeneous. As such the software products may slightly differ by their design and in addition the prerequisites (e.g., existence of root exploit) to succeed an attack may greatly vary from one device to another. The direct consequence is that it is no necessary easy to replicate an attack identified on a particular environment (i.e., device and operating system); onto another environment (i.e., different device or different instance of the software). For example installed software and related sensitive data could be bound to the device/instance to reduce replicability (here cloning) opportunities.

To represent the capability of an attacker to exploit the vulnerability on a large set of devices or with limited customization effort, a new factor "Replicability" is defined as follows:

- Easy: the attack is generic and thus can only be exploited on all or several range of instances of the software products and/or are conditioned to the presence of technical prerequisites that are commonly encountered within the (mobile) environment including on last generation of devices
- Moderate: the attack can only be exploited on a full range of instances of the software products and/or are conditioned to the presence of technical prerequisites that are commonly encountered within the (mobile) environment but on limited number of devices (e.g., outdated or more than 3 years old smartphones)
- Difficult: the attack can only be exploited on very few instances of the software products and/or are conditioned to the presence of technical prerequisites that are very unlikely within the mobile environment

As illustrated above, the “Replicability” factor shall strictly focus on the likelihood of the attack replication in several contexts: is it possible to repeat the attack and what are the difficulties to setup the attack on various software products instances? It shall not be confused however with the impact of the attack which are use cases specific and therefore shall be considered during security risk assessment but not during attack potential calculation.

The following table gives a replicability level according to the likelihood of the attack.

Attack likelihood	Environment	Replicability
The attack setup in exploitation is strictly identical in several contexts.	For a large set of operating system versions (e.g., Android from 2.0 to 7) and a large set of devices.	Easy
The attack mandates existence of device or mobile OS vulnerabilities allowing elevated attacker privileges and illegal accesses to OS core features.	Privilege escalation (e.g., jailbreak, root exploit) or OS vulnerabilities for some years old devices or OS (e.g., 3 years old).	Moderate
The attack only works on a given version of the environment (device and OS) and setup requires to be deeply customized for a specific software instance (e.g., because of existing protections).	Exploitable for a given device and OS version (e.g., Windows Mobile).	Difficult

6. Calculation of attack potential

Thanks to the previous chapter describing the relevant factors, the following table makes a correspondence between a factor and a numeric value, to enable a global rating of the attack. This table is inspired from [ISOIEC-18045].

Factor	Value for identification phase	Value for exploitation phase
Elapsed time		
<= one hour	0	0
<= one day	1	2
<= one week	2	4
<= one month	3	6
> one month	5	8
Expertise		
Layman	0	0
Proficient	2	3

Expert	5	5
Multiple Expert	7	7
Knowledge of TOE		
Public	0	0
Restricted	2	2
Sensitive	4	4
Critical	6	6
Window of opportunity		
Unlimited access	0	0
Easy	1	1
Moderate	2	3
Difficult	4	5
Equipment		
Standard	0	0
Specialized	1	3
Bespoke	2	5
Replicability		
Easy	NA	0
Moderate	NA	3
Difficult	NA	6

That makes software products meeting FIDO requirements at L1+ from rating 26 (and L1 from rating 12):

Attack potential value (sum)	TOE resistant to attackers with attack potential of
0-11	No rating
12-25	Basic
26-32	Moderate
>32	High

7. Examples of software attack methods

Security mechanisms and countermeasures ensuring the protection of software L1+ products may involve several technologies and underlying mechanisms.

To ease the work of laboratories during FIDO certification and to allow them to individually test the resistance of some mechanisms (e.g., WBC), it may be requested to developers to provide specific FIDO product configuration (e.g., with some protection deactivated) to proceed with the tests.

If such configuration has been used and vulnerabilities are identified, laboratories may make a partial attack rating but with the following constraints:

- The specific configuration shall be reflected in factor “Window of opportunity” and any additional information (e.g., specification, source code) in the “Knowledge of the TOE” factor.

In addition for each partial attack, if an example of rating is computed, it shall only be used to give an indication on the necessary efforts to perform a partial attack. Consequently the number of given points (total sum) shall not be computed/reported at all since it is only relevant to a full attack path.

The partial attacks described in this chapter only target a specific technology and are therefore not sufficient to demonstrate exploitable vulnerabilities. A full attack path in the context of software L1+ products likely needs to combine several partial attacks (including associated prerequisites).

The TOE is the binary application package which can be downloaded from the application store, as the most widespread use case in mobile environment, which is the L1+ target.

The following list of attacks is detailed in [CYBER]. All the following attacks are SOFTWARE attacks.

7.1 Partial attacks on the code protections

The focus is made on reverse engineering and runtime data extraction (tracing).

Reverse engineering covers a wide range of techniques and tools. One of the aims is to give the attacker a better understanding of the purpose of a binary or a function, often to come up with an equivalent but simplified code. It is of particular interest when targeting undocumented features.

Reverse engineering is attacker’s ability to retrieve the code behavior (at design level) from software binary code or source code.

Reverse engineering tasks may be decomposed in several steps:

- First step consists into transforming (disassembling or decompiling) the binary code into source code equivalent. Ability of such transformation may require standard or specific equipment that could be automated. Similarly, the required expertise depends on code size/complexity and implemented countermeasures (see obfuscation techniques in [CYBER]).
- Such primary transformation may not be enough to have an attacker’s useful understanding of the software behavior (at design level) and thus additional transformation or code modelization (including cryptanalysis for White box Cryptography) may be required with different equipment and expertise.

Tracing, or the ability to obtain a trace of a software execution (record about a software execution or

usage of resources …) is key for cryptographic attacks (side-channel attacks in particular). In the context of this document tracing is more generally linked to dynamic (runtime) analysis of the application's behavior, which is also of great interest to complete reverse engineering task. Tracing can be achieved at multiple levels (memory accesses, instruction registers), through multiple tools (debugger, Dynamic Binary Instrumentation (DBI), emulator).

Tracing tasks may be protected by several code protection mechanisms as exposed in [CYBER]. As such, the software could implement mechanisms to detect tampering, debugging or execution in emulated environment which are required to allow monitoring of the software runtime execution.

7.2 Partial attacks on the crypto protections (WBC)

The focus is made on Differential Computation Analysis and Differential Fault Analysis.

Similarly to other technical domains, the presence of a side-channel may be exploited to disclose data manipulated by a software. While physical properties (ex: power consumption or electro-magnetic emanations during software execution) can still be used, the white box attack context requires to also consider software manipulations. As such, classical side-channel analysis techniques could be applied to software traces of a binary execution and therefore tracing tasks, as reported in previous section, is a prerequisite.

There are various examples of side-channel techniques. For each of the side-channel techniques, additional prerequisites may apply for the effectiveness of the attack such as ability to choose (i.e., controlled by attacker) the input to the cryptographic engine.

Similarly to other technical domains, the execution of a software could be perturbed (permanently or temporarily) by some faults creating an abnormal behavior that could be further exploited for controlling software execution or disclosing software data. For smart cards domains such faults could be generated by physical resources (ex: power, laser, electro-magnetic glitches …). In the context of FIDO L1+ products only faults generated by software are to be considered such as manipulation of registers (ex: Program Counter), memory or instruction flow.

Computation of cryptographic algorithms (including WBC) is a particular target possibly showing algebraic weaknesses (ex: Differential Fault Analysis (DFA) on AES).

7.3 Relevant (full) attacks methods

This section describes two attacks (one local, one remote) against a FIDO L1+ Authenticator.

7.3.1 Local Expert attacker

Scope of the attack

Key (Authentication) disclosure through local leakage (ECDSA key extraction through DCA) accessing locally to user device and reusing disclosed sensitive data in attacker controlled device.

Associated threat to the attack

Considering a threat as, attacker masquerades user using genuine authentication method and the authentication key discovered from the user device and installed in attacker device.

Description of the attack

Identification

1. Attacker is supposed to be able to install application on its own device and to be registered to be able to run application on its own device. Then attacker can try to understand and then defeat the software protection of the application and underneath environment (i.e., OS).
2. Attacker has obtained knowledge of application own device and will try to demonstrate its ability to retrieve user sensitive data from user device. As a result, a procedure (automated partially or not) to retrieve sensitive data of application's instance on user device is implemented (this scenario requires direct local access to the user device).
3. Attacker that captured user sensitive data using above implemented procedure will demonstrate its ability to reuse (clone) user sensitive data in attacker device, thus masquerading the user. As a result, a procedure (automated partially or not) to perform illegal authentication with imported user sensitive data is implemented.

Exploitation

1. Attacker accesses to user device and applies procedure defined in identification steps to capture sensitive user data.
2. Attacker applies procedure defined in identification steps to perform authentication operation on behalf of user and usits own device.

7.3.2 Remote Expert attacker

Scope of the attack

Key (Authentication) disclosure through remote leakage (ECDSA key extraction through DCA) accessing remotely to user device and reusing disclosed sensitive data in attacker device.

Associated threat to the attack

Considering a threat as, attacker masquerades user using genuine authentication method and the authentication key discovered from the user device using remote attack and installed in attacker device.

Description of the attack

Identification

1. Same as for local attacker.
2. Attacker has obtained knowledge of application on its own device and will try to retrieve user sensitive data from user device by installing and executing some malicious code on targeted device remotely. As a result a procedure (automated partially or not) to retrieve sensitive data of application instance on user device is implemented.
3. Attacker will try to demonstrate its ability to reuse user sensitive data in attacker device to masquerade user from attacker device. As a result a procedure (automated partially or not) to perform authentication with disclosed user sensitive data using application instance on the attacker device.

Exploitation

1. Attacker accesses to user device and applies procedure defined in identification steps to capture sensitive user data.
2. Attacker applies procedure defined in identification steps to perform authentication operation on behalf of user and using its own device.

Major difference is ability to address several user devices with the same remote procedure, thus impacting replicability factor.

A. References

A.1 Normative references

[AttackPotentialSmartcards]
Application of Attack Potential to Smartcards. January 2019. URL: https://www.sogis.eu/documents/cc/domains/sc/JIL-Application-of-Attack-Potential-to-Smartcards-v3-0.pdf

[CYBER]
ETSI TR 103 642 CYBER; Security techniques for protecting software in a white box model. October 2018. URL: https://www.etsi.org/deliver/etsi_tr/103600_103699/103642/01.01.01_60/tr_103642v010101p.pdf

[ISOIEC-18045]
Information technology -- Security techniques -- Methodology for IT security evaluation. August 2008. URL: https://www.iso.org/standard/46412.html

[TEE-PP]
. *GPD_SPE_021 TEE Protection Profile version 1.3*. September 2020. URL: https://globalplatform.org/specs-library/tee-protection-profile-v1-3/