Games based on active NFC objects : model and security requirements

Florent Fortat, Maryline Laurent, Michel Simatic

Games based on active NFC Objects: Model and Security Requirements

Florent Fortat*†, Maryline Laurent†, Michel Simatic†
*Hanakai Studio, Paris, France
†SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay, EVRY, France
{florent.fortat, maryline.laurent, michel.simatic}@telecom-sudparis.eu

***Abstract*—Cheating in video games is a critical financial matter for game developers. With games now integrating physical objects through NFC, new cheating techniques have emerged, including characteristic boosting of the objects, duplication of objects and introduction of new unauthorized objects. In this paper, we address this problem for games based on active NFC objects. Having active objects in a game allows for new possibilities of interaction yet to be seen, including offline interactions between objects outside of the game. We identify four top security requirements for these games to remain resistant to cheating activities. This analysis is original as it introduces a new model with active NFC objects. Our system is composed of a server, a device (computer, console or smartphone) equipped with an NFC reader, and NFC objects. We perform a risk analysis to identify the weak points of this base system. We list several realistic attacks exploiting the system's weaknesses. Finally, to address them, we design four cost effective security requirements.**

I. Introduction

Games are made to entertain players but some players prefer to break their rules by cheating, either to experience the game in a different way, to vandalize it or to dominate other players in online games [1], [2]. Recently, games using physical objects appeared and are the target of new cheating techniques. For instance, Brandon Wilson documented the protocol used for Skylander's NFC (Near Field Communication [3]) transceiver by reverse engineering its driver [4]. It led to the creation of a tool to modify the figurines' data [5]. Another example using a third party device is Datel's PowerSave to boost the statistics of Amiibo figurines [6].

In a near future, this kind of games is going to experience a major evolution by giving more computational power to its physical objects, as well as their own power source. Thanks to this evolution, objects will be able to be operated independently of any gaming system. Such evolution enables players to interact with other players with their respective objects outside of the game. It brings the gaming experience to a whole new level for players, beyond what is currently offered with NFC objects like Skylanders, Disney Infinity, Amiibo and Lego Dimensions. But such experience will bring new possibilities to cheat. Hopefully, the increase in computational power is an opportunity to set up new security measures to prevent these cheating flaws.

In this paper, we present a framework to secure games based on active NFC objects against cheaters. This framework takes into account cost efficiency, and several realistic capabilities of possible attackers. After first proposing our modeling of the game system, we perform a thorough risk analysis which helps identifying the top four security requirements for achieving a rational security level. Finally, the security requirements are discussed with regard to the cost efficiency, and some conceivable attackers capabilities.

The rest of the paper is organized as follows: Section II presents the model of our system. Section III provides a risk analysis based on the previous model. Section IV identifies top four realistic security requirements to secure the system. Section V concludes.

This work is part of the French National Research Agency (ANR) project INCOME (anr-income.fr) (ANR-11-INFR-009, 2012-2015)

II. Model

In this section, we define a model for active NFC objects used in a gaming context. The goal of this model is to formalize the characteristics of such objects. This formalization provides a framework to be used for future work regarding active NFC objects in gaming and to evaluate cheating and security issues.

A. EBIOS

For this risk analysis, we apply EBIOS (Expression of Needs and Identification of Security Objectives) method to our model. EBIOS is a tool to audit security in information systems [7]. Even though we use this tool, for the sake of clarity, we use an alternative vocabulary for some terms to better match real gaming world problems and solutions. For the same reasons, we also use sub-concepts as main concepts in this paper.

Our model is composed of two EBIOS concepts : *supporting assets* which are base elements like hardware, software, people, etc., and *primary assets* which are functional elements relying on *supporting assets*. In our model, we respectively call them components and tasks as those terms better reflect the elements of our system.

Regarding the risk analysis, people able to harm the system are *threat sources*. We call them attackers. For the analysis we have two concepts: threat scenarios which describe the threats on the system and *feared events* which describe what impact a given event would have if it happened. In this article, we only keep the impacts of the feared events as they are more understandable in our gaming context. A threat is associated with a component and has a likelihood level. An impact is associated with a task and has a gravity level. By associating

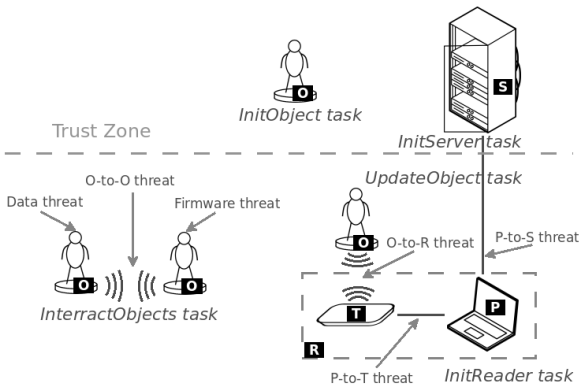

Fig. 1. Model and threats for games based on active NFC objects

tasks with the components they use, we have a list of risks characterized by the combination of a threat and an impact. In this paper, we choose to translate these risks into concrete actions we call attacks.

B. Components

In our model, we consider three components (see Figure 1): the NFC object, the reader and the server. In this paper, unlike with other models [8], [9], we separate the server from the reader because they are not operated by the same people. In the gaming context, the most general configuration is to have a central server managed by the game developer and a client software running on the gamers' devices. In the context of gaming with NFC objects, the gamers' devices are equipped with an NFC transceiver.

An NFC object (O in Figure 1) can either be an active NFC object or an object with a simple passive NFC tag that anyone can read and write data on. An active NFC object is battery-powered and has a computational power similar to a smartcard. An object can communicate with another if at least one of the two is active. It can only communicate with one other device at a time. Finally, multiple types of object can exist for a given game (e.g. different characters).

A reader (R in Figure 1) is an NFC transceiver (T in Figure 1) connected to a computing device. The reader can be a smartphone as well as a computer or a gaming console with an NFC board connected to it. The transceiver communicates with a program (P in Figure 1) running on the device. The transceiver has no knowledge of the legitimacy of the program. The reader can communicate with one or more objects at a time depending on the design of the transceiver.

The server (S in Figure 1) is a program running on a backend system owned and controlled by the game developer. It stores a copy of all the NFC objects' data. The NFC objects interact with the server through the readers when those can communicate with the server. It can have multiple communication channels running at a time.

C. Data and firmware

We mentioned in section II-B that NFC objects can store data. We have two types of data in both active and passive objects as well as a firmware in active objects.

First, we have the identifier. Every object has a unique ID attributed to itself as described by every NFC standards. Then, there is the object's data. Its format is designed by the game developer and it stores the state of what represents the object. It is a key part to enable interaction between objects. Finally, active objects have a firmware, that is, the program running on these objects. It contains the algorithm used to determine how to interact with other objects.

D. Operational tasks

This section defines our framework based on the components presented in section II-B. Below are listed tasks known as the EBIOS *primary assets* for our system. We specify which components are used to perform each task.

InitializeServer(server): This task initializes the server. It creates an empty database to store the data of the NFC objects. It is invoked by the game developer when the system is going to its production phase.

InitializeReader(reader): This task initializes the program on the user's device. It is invoked during the installation of the application by the user.

InitializeObject(object, server): This task initializes an object. It sets the object with a unique ID and its initial data which is sent to the server to be stored. It is invoked during the manufacturing of the object at the factory.

UpdateObject(object, reader, server): This is invoked by the reader when the game must modify the data of the object. Thanks to an eponymous communication protocol, the server checks the object's data and updates or corrects it as needed and then sends it back to the object.

InteractObjects(object1, object2): This task is triggered when two objects can communicate via NFC. Thanks to an eponymous communication protocol, they share their data to each other and determine an action to perform.

III. Risk analysis

We identify two kinds of attackers in our model: the cheater and the counterfeiter. The **cheater** is a player motivated by ingame contents, like improving his character's statistics or accessing locked skills for example. We consider every cheater equally, as cheating tools can be made available to cheaters with a small skillset. The **counterfeiter** is an organization motivated by making money through the selling of unauthorized goods, tools or services. It has an extended skillset to attack the system and huge resources supported by potentially illegal activities.

Our model, assumes different **levels of trust** on the three components. The server is fully trusted. The reader cannot be trusted as it can be compromised. The NFC object cannot be trusted either. A passive object can be compromised by any reader and can be cloned. Moreover an active object can be counterfeited.

Based on our three components, we identify six threats in our model (see figure 1):

Data modification: On passive NFC objects, the data stored on the tag can be altered or duplicated to other tags.

Firmware retrieval: On active NFC objects, the firmware and the data can be retrieved and modified to create unauthorized active NFC objects.

Object to object (O-to-O) eavesdropping: All the data exchanged during the execution of the *InteractObjects* task

can be retrieved, exposing the protocol and the object's data.
Object to reader (O-to-R) eavesdropping: All the data exchanged during the execution of the *UpdateObject* task can be retrieved, exposing the protocol and the object's data.
Program to transceiver man in the middle (P-to-T MITM): An attacker can create a program to act as a man in the middle to alter data transferring between the device and the transceiver by reverse engineering the software on the reader's device.
Program to server man in the middle (P-to-S MITM): An attacker can create a program to act as a man in the middle to alter data transferring between the device and the server by reverse engineering the software on the reader's device.

Based on those threats, we identify five risks. We then translate them into the following four attacks. The first three attacks can be used by tools to facilitate cheating. The fourth attack, is related to counterfeiting figurines.
Boost characteristics on object: An attacker can modify characteristics on an object using the data modification threat, the P-to-T MITM threat or the P-to-S MITM threat.
Boost characteristics on reader: Characteristics can be boosted using the P-to-T MITM threat before the program receives them. They are then reverted to their original state using the P-to-S MITM threat before being sent to the server.
Fake action: Using a custom reader, an attacker can perform illegitimate actions on an object. This attack uses the O-to-O eavesdropping threat to retrieve the protocol.
Object counterfeiting: Passive objects can be counterfeited using the data modification threat to copy to a new tag. Active objects can be counterfeited using firmware retrieval threat or by combining the O-to-O and O-to-R eavesdropping threats. Enabling either to retrieve the firmware as a binary file or to create a new firmware using the same protocols.

IV. Requirements and discussion

In this section, we identify the following top four security requirements addressing different parts of the four attacks set forth in section III. To better understand how our four security requirements secure the NFC-based game model we present, Figure 2 summarizes the relationship between threats, attacks and requirements.
Object to server secure communication: It ensures the data will not be altered when saved on the server. It also prevents the easiest way for an attacker from retrieving the object's data and to study the *UpdateObject* protocol. It requires a one time software development on both sides. This requirement implies that the microcontroller must be able to compute cryptographic primitives. But, this is acceptable because we can use a cheap enough microcontroller.
Object to object light secure communication: It prevents an attacker from replaying the session on another object. It also prevents the easiest way for an attacker from studying the *InteractObjects* protocol. The software development required here is shared with the previous security requirement. So it does not bring any additional cost.
Sign data: When the object is updated, the server signs the data and sends the signature along with the data. When the object is read by an active object or a reader, they can check if the current data have been approved by the server. This prevents the use of altered data. It requires a one time software development on the server side.
Mandatory online checks: By forcing the program on the reader to check the object's data with the server when the object interacts with the reader, we prevent counterfeited objects from being introduced into the system. It requires a one time software development on the reader and the server.

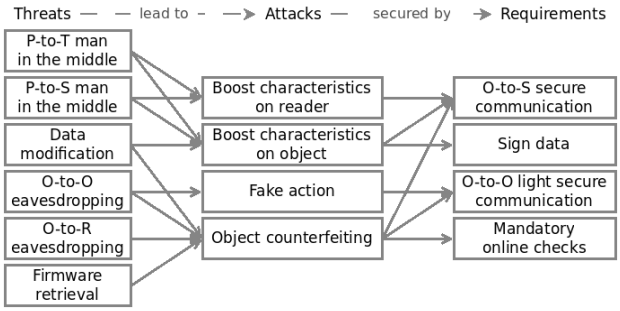

Fig. 2. Relationship between threats, attacks and requirements.

V. Conclusion

Thanks to a thorough risk analysis, this paper identifies top four realistic and cost efficient security requirements for securing games based on active NFC objects against cheaters: secure communications between the object and the server as well as between two objects, sign data stored on objects and perform regular mandatory online checks of the objects. The objective of these requirements is to help game developers to protect their games from players boosting the characteristics of their objects and playing with counterfeited objects.

Games based on active NFC objects are yet to be seen. They have the potential to bring gaming experience to a whole new level with the addition of out of the game interactions which will still have an impact in the game via the players active objects. By preventing new cheat techniques and counterfeiting, the requirements and the implementation directions presented in this paper help protecting both the players' community of the game and the business model of the game company for promising new game experiences.

References

[1] J. Smed and H. Hakonen, "Cheating Prevention," in *Algorithms and Networking for Computer Games*. John Wiley & Sons, Ltd, 2006, pp. 213–225.

[2] M. Consalvo, *Cheating: gaining advantage in videogames*. Cambridge, Mass: MIT Press, 2007.

[3] NFCForum, "NFC Forum," 2015. http://nfc-forum.org/

[4] B. Wilson, "skylanders portal documentation," Oct. 2011. http://brandonw.net/

[5] B. Wilson, "setting the record straight," Mar. 2014. http://brandonw.net/

[6] Codejunkies, "PowerSaves for Amiibo," 2015. http://uk.codejunkies.com/Products/PowerSaves-for-Amiibo__EF001187.aspx

[7] ANSSI, "EBIOS 2010 - Expression of Needs and Identification of Security Objectives," 2010. http://www.ssi.gouv.fr/en/the-anssi/publications-109/methods-to-achieve-iss/ebios-2010-expression-of-needs-and-identification-of-security-objectives.html

[8] S. Vaudenay, "On privacy models for RFID," in *Advances in Cryptology–ASIACRYPT 2007*. Springer, 2007, pp. 68–87.

[9] J. Hermans, A. Pashalidis, F. Vercauteren, and B. Preneel, "A new RFID privacy model," in *Computer Security–ESORICS 2011*. Springer, 2011, pp. 568–587.