A Survey of Security Analysis in Federated Identity Management

Sean Simpson and Thomas Groß

Newcastle University

Abstract. *We conduct a survey of security analysis in Federated Identity Management (FIM). We use the Malicious and Accidental Fault Tolerance framework (MAFTIA) to categorise security incidents in FIM. When security incidents are categorised using MAFTIA we can paint a picture of the landscape of problems that have been studied in FIM. We state common failure paths of FIM systems and how FIM systems can be protected from those failures.*

Keywords: FIM, Survey, Dependability, MAFTIA, Microsoft Passport, OAuth, OpenID, Facebook Connect, SAML, Liberty Alliance

1 Introduction

Web applications typically employ a username/password authentication structure that forms a user's credentials to access the service. The problem that many modern users face is that in using different websites the user has to decide if they want to use the same credentials they used for another website or use different credentials.

Federated Identity Management (FIM) aims to alleviate this problem by allowing a user to sign into multiple Service Providers (SP) using the same credentials. How FIM solutions do this is by using three-way authentication. In three-way authentication, a user will, as a prerequisite, register with an Identity Provider (IdP) that will serve as a central figure to authenticate the user. Typically in FIM, the user will attempt to access a SP which will then redirect the user to authenticate with the IdP which will vouch for the user, communicating with the SP to say that the user is who they say they are. FIM solutions are seeing increasing use. One only has to look at the prevalence and ubiquitous nature of an Identity Provider like Facebook Connect to see that for a lot of users, Federated Identity Management exists in their life and they dont even know it [7]. Many attacks on protocols used in FIM have been found, which is motivation to do a survey in the area.

Protocols in FIM have been analysed by others in an attempt to find problems. The papers presented from the analysis find a vulnerability in a system, and show how that vulnerability is attacked, which eventually causes a security failure. The problem is that there is a lot of different information on the analysis of security protocols in FIM that remains uncompiled. Our goal is to create a survey paper for security analysis in FIM. We review existing peer reveiwed academic publications that perform security analysis on FIM protocols to establish a common ground and collect knowledge. In addition, we want to create a unified way of looking at security incidents in FIM and offer a framework to do that.

2 Methodology

This section details the concepts we use to categorise security incidents in FIM. A security incident is not some atomic process that cannot be broken down. In fact, a security incident has vulnerabilities, attacks, intrusions, errors and eventual failures. Categorising security incidents is not some new concept but is something that has been investigated by many others. One Particular way of categorising security incidents is by using concepts from the Malicious and Accidental Fault Tolerance (MAFTIA) project [17] . The MAFTIA project itself uses concepts derived from the area of fault tolerance and dependability. Although we are not practicing fault tolerance, dependability offers a well established way of looking at systems. Dependability concepts and works based on dependability concepts (such as the MAFTIA project), introduce a number of different terms that we use. We will give a brief overview of relevant dependability and MAFTIA concepts.

2.1 Dependability Concepts

The need for dependability on systems emerged because of an increasing reliance on systems in the modern era. Before the practice of fault tolerance and dependability, system designers were mostly concerned in stopping a system from failing by ensuring faults do not exist in the system (fault prevention). Fault Tolerance and dependability, as an additional step to fault prevention, accepts that a system is not perfect and tries to tolerate a fault in such a way to not cause a system failure. The original work that defined dependability concepts is presented by Avizienis, Laprie and Randell [2].

Definition 1 (System). *A* system *is a mechanism which behaves in a way according to some design. The design of a system is envisioned from a specification of what the system should do. The specification of a system can be formally written down or just exist as an idea. A system interacts with systems around it through a system boundary.*

Definition 2 (Fault, Error, Failure).

Fault The hypothesized and the adjudged cause of an error.

Error The part in a system state that may lead to a failure.

Failure When the system is adjudged to not be offering correct service.

2.2 MAFTIA Concepts

Before the MAFTIA project took place the area of dependability was mainly concerned with accidental faults in a system. The MAFTIA project took a different view point and changed the model to look at system reliability with a malicious adversary factored in. The main differences between MAFTIA concepts and more general dependability concepts is that the notion of a fault now needs to accommodate for malicious behaviours. In MAFTIA, a fault is broken down into three seperate elements: vulnerability, attack and intrusion which are defined below.

Definition 3. Vulnerability *A fault that is created during the development or operation of the system that if exploited causes an intrusion.*

Attack *A malicious interaction fault that attempts to exploit a vulnerability. Can be thought of as an intrusion attempt.*

Intrusion *An adversary-introduced fault. An intrusion is created as the product from an attack successfully exploiting a vulnerability by an adversary.*

2.3 Attack Class

In addition to the framework provided by others, we introduce some of our own notions to assist us in practically using the dependability/MAFTIA taxonomy to categorise security incidents in FIM. The first of these is an attack class.

Attack Class: The Attack Class abstraction is our adaption of the original MAFTIA project concepts. The attack class abstraction contains any intrusions or errors that happen as a result of any attack actions. We have observed that once a vulnerability is attacked a set of states and actions occur in a system which is similar across different papers. We capture the attacks, intrusions and errors that occur in a security incident in an abstraction known as the *attack class*. A definition and example is given below.

Definition 4. *An* attack class *is a collection of attack events, intrusions and resultant errors in a system. If the errors generated from the malicious faults are not dealt with, a failure could follow.*

Attack Class for a bogus merchant: One can think of all of these different concepts being either states or events. A vulnerability is a state which exists in the system. An attack is an event caused by an adversary to exploit a vulnerability. An intrusion is the resultant state of a successful attack on a vulnerability. An error is an erroneous state in the system which occurs as a result of a fault (in this case being the vulnerability, attack, intrusion combination). In a bogus merchant attack, the act of setting up the bogus merchant is an attack while the intrusion state is the bogus merchant existing. When a user browses to the bogus merchant and types in valid user credentials an error occurs because the adversary now has access to secret user credentials which the adversary can use to log in to the legitimate IdP in the guise of the user. The act of the adversary logging on using stolen user credentials is technically an attack since an attack is defined as any interactive step an adversary takes. The adversary logging on to the legitimate IdP in the guise of the user causes another error state and puts the system in a position to fail because the adversary now has access to the user's account.

The above example shows how an attack class can be used to capture the attacks, intrusions and errors that occur when a vulnerability is exploited in a system and up to the point in which the system is about to fail. We therefore use the attack class abstraction to categorise the chain of events and states an adversary sets in motion to cause a failure. Using the concepts of a vulnerability, attack class and a failure we can more clearly define a security incident in FIM.

2.4 FIM System Boundary

Since the work we are producing is a survey paper, we want to include as much information about the security incidents that occur in FIM as possible. The problem is that vulnerabilities can occur outside of a FIM system boundary. An example of such a vulnerability is a user mistake, where a user does something like not check a certificate for a site, doesn't check the site URL etc. We want to include vulnerabilities such as this in our findings, but we also want to specifically state that these faults occur outside the FIM system boundary. We do believe that systems should be secure despite the environment around them but it is also useful to capture the issues that do exist outside of that system. To state whether or not a vulnerability occurs inside the system or not, we first need to define what we consider inside and outside of the system.

The components of the system which are inside the system: User's browser, Service Providers (of which there can be any number), IdP (of which there can be any number, but the majority of time the system is modelled as having a single IdP), any communication channels that are used for any of these components to communicate and the protocol flow itself. We do not exhaustively list the components that are considered outside of the system but examples: User, DNS and CAs.

2.5 User Trust Requirements in FIM

The first thing to establish is what we mean by trust. We use the definition from [13]:

Definition 5 (Trust). *Trust of a party A to a party B for a service X is the measurable belief of A in that B behaves dependably for a specified period within a specified context (in relation to service X).*

There have also been attempts to formulate trust in FIM. For instance, [10] models trust from the point of view of every party that is invovled in a FIM system. The issue with these requirements is that it is not clear how many of these requirements need to be broken before a system fails. Another solution is by FutureID where a very detailed list of requirements is laid out for FIM system. The problem with these requirements are the same in that it's not clear how many of those requirements need to be broken before the system fails. In addition, these requirements are also too low level for our purposes, for instance, one security requirement states that the IdP locks itself to a user after three login attempts. These requirements are too low level for our purposes.

We attempt to improve on these trust requirements in FIM by showing when it is that a FIM system fails and what are the components that caused that failure. To do this, we use our system boundary approach defined in this section to view a FIM system as a system of systems. A user interacts with the FIM system to authenticate themself for some purpose. During this interaction, the user expects confidentiality of their information, integritiy of the actions they perform, and the availability of the FIM service.

If the CIA requirements of the user are broken, from the user's point of view, the system has failed. There are a number of other external actors outside a FIM system boundary such as businesses, server admins, and governments which could place different requirements on a FIM system. For example, a businessman might require a FIM system to use as little bandwidth as possible in order to minimise costs but such a requirement has little effect on the user. We do not attempt to exhaustively list all of the requirements that various other actors might place on a FIM system. For this reason, it could be the case that some actor judges the FIM system to have failed, while from the user's point of view, it hasn't. We consider failures of a FIM system from a user's point of view to simplify how we view FIM systems.

The main components we consider in a FIM system are the user's browser, SPs, and IdPs. Any one of these can fail in various ways but they are considered a subsystem of the FIM system. In the next subsection, we consider what it means for a subsystem to fail. When a subsystem of a FIM system fails that does not necessarily mean that a user's confidentiality, integrity or access to the system is compromised so the system has not yet failed. It is when one, or a combination of subsystems fail in such a way that the user's confidentiality, integrity or availability requirements are compromised that a FIM system is said to fail.

Using the concept of Trust requirements in FIM, we can establish the point of failure of a FIM system from a user's point of view and the parties in a FIM system responsible for that failure.

2.6 FIM trust requirements on subsystems

In the previous subsection we discuss the trust that a user puts on a FIM system. If the FIM system breaks that trust, then from the user's perspective the FIM system has failed. In the same way that when a FIM system fails it is because it doesn't meet the trust requirements placed on it by a user, a FIM subsystem fails when it doesn't meet the trust requirements placed on it by a FIM system.

IdPs, SPs and browsers are expected to follow the protocol specification and keep any sensitive information confidential. Confidential data includes username/password credentials and user profile data. The browser should also detect fraudulent websites.

2.7 Process of Extracting Vulnerabilities, Attack classes and Failures from Existing Work

In our work we survey a number of other papers and in these papers security incidents found by the author are showcased. It is important that the process of identifying vulnerabilities, attack classes and failures from existing work is sound.

It is often the case that specific vulnerabilities, attack classes and failures are not identified explicitly. If this is the case, we will claim that the specific vulnerability, attack class or failure not explicitly stated has happened and call it implicit. We use different studies to backup our claim where similar security incidents have happened and in those similar studies it is pointed out that the specific vulnerability, attack class or failure has happened. The reasoning behind this is, for example,to find a vulnerability within the system boundary that causes an attack to be possible. For instance, simply stating that in a bogus merchant attack the user is at fault is not sufficient in trying to fix the vulnerability since a system designer cannot change the user; this is why we model the user as a component outside the system boundary.

In some studies it is the case that the authors engineered an attack on a FIM system and executed the attack on real implementations. Other authors conceptualise an attack but don't execute it in practice. Regardless of if an attack is conceptual or if it is executed in practice, we include it in our survey. Completeness of the survey is the reason for including attacks both conceptual and concrete. However, there are instances in the papers we've surveyed where authors list vulnerabilities and risks of the protocol being analysed but don't state how those vulnerabilities are exploited. For the full security incident to be included in our survey the way a vulnerability is exploited must be included.

Identifying Vulnerabilities in FIM: Where the authors identify the part of a system state that is adjudged to make that attack possible, we identify that part of the system as a vulnerability.

Identifying Attack Classes: Attack classes consist of attack events, which create intrusion states, which cause error states in a system. Attack classes can be identified through the total set of actions and states that exist in the target system after vulnerabilities have been exploited through an attack and before a failure occurs.

Identifying Failures: A failure is the end goal for an adversary. The papers we analyse focus on perserving user confidentiality and integrity and so when that is breached, typically through an adversary impersonating the user, a failure is indentified. If a user's profile data is deemed to be exposed then that is regarded as a failure, this can be seen when an adversary attempts to spoof a user's data. There has also been discussion on attacks that focus on disrupting the availability of FIM systems. If it is thought that a FIM system can fail through inadequate availability, that is also a failure. In addition, we identify component failures as the set of components that fail in order for the CIA failure to occur.

2.8 Motivation for Using the Dependability/MAFTIA Framework:

We have given an overview of Dependability, MAFTIA, and our own notions but what is the purpose of using it? Authors approach analysis of security protocols in different ways and the main concern is presenting a unique contribution that undermines the security of the protocol. Authors may put special emphasis on vulnerabilities that have previously been undiscovered or on more complex attacks that allow a failure to take place. Our goal is to digest the security analysis produced by others and present the findings from those reports in a uniform way which we can do using MAFTIA.

One reason why MAFTIA can help in presenting results in a uniform way is through assignment and classification of incidents in FIM. When reading the reports by other authors, we assign the specific problem they describe to a vulnerability, attack class or failure based on the information provided by the author and also by cross-checking that information to other sources to ensure that the information is sound. The fundamentals of how we assign a specific problem to a vulnerability, attack class or failure is described in section 2.7. When a problem has been assigned, it needs to be classified. For instance, we identify a vulnerability called unencrypted communications. Information on how we assign vulnerabilities and the other MAFTIA concepts can also be found in 2. The authors who present an attack might express that unencrypted communications has taken place in a slightly different way so when we disgest that authors paper it is important to decide if a problem presented is a new classification or can be safely lumped into an existing classification. Classifications allow us to talk about the different problems presented by different papers generally. In a rather simple example of us talking generally about a problem, we can propose that a defence for all unencrypted communications would be to encrypt the communications. If the problems were not classified we would have to refer to them on a case-by-case basis which could get repetitive and dilute the important details.

Dependability is a topic which has been studied for sometime. So far we have talked about putting problems into dependability terms but have ignored what the purpose of dependability is in the first place. Dependability is used to create fault tolerance in systems. In the future, we could use fault tolerance techniques to reason about problems with FIM.

2.9 Survey Completeness

How do we know that our survey covers the differernt works on the security analysis of FIM protocols for all the different FIM technologies to a resonable level. We will first show how our survey considers the relevant FIM technologies and then explain our methodology for ensuring we have a complete picture of the security analysis that exist for those technologies.
Ensuring a complete picture on technology: There has been work on surveying the technology used in FIM systems [6]. We use this as a baseline for considered FIM protocols.

Ensuring a complete picture on analysis: For each security protocol considered there is usually a standout paper that contributes an analysis for that paper. This standout paper can be identified by the traction and citation count it has garnered since publication. The standout paper will build on other analysis on the protocol or itself be cited by others. From looking at these citations, a web of knowledge can be built about the work that exists for a protocol. Take it another step further and then look at the citations for a particular paper that has either cited or been cited by the standout paper. Continue this process until a complete map has been built of the works that exist for a particular protocol. This methodology carries the risk of papers that have been produced in isolation are not found so additional searches have been carried out using search engines like google scholar but no isolated papers were found in this way.

3 Papers Analysed

Providing a detailed description of the analysis of all the papers we looked at is quite lengthly. This sections provides an overview of the papers we looked at but for more detail, refer to our technical report for more details [19]. This report includes more precise details on what vulnerabilities, attack classes, component failures and CIA failures were identified and where.

3.1 Microsoft Passport

Risks of the passport single singon protocol[11]: Microsoft Passport was one of the first attempts at FIM. At the time the paper was written the concept of FIM was a new one and the paper was one of the first papers to crittically examine FIM. As such, in addition to concrete problems with FIM that are brought to light, risks associated with FIM in general are also pointed out. In particular, the practical attacks focused on in this paper are bogus merchant attacks, DNS attacks and DoS attacks. The paper talks about risks such as a central point of attack in Passport; however, there is no reason that this central point of attack is limited to only Passport as many other FIM protocols rely on a single IdP (e.g. Facebook Connect). Additionally, some concern is expressed regarding the use of cookies in Microsoft Passport in the event public machines are used.

Other Microsoft Passport Contributions: Oppliger [14] Outlines a number of vulnerabilities in Microsoft Passport. For instance: reliance on DNS, Key Management and the existence of a single point of failure. In addition discussion of the issues pointed out by Korman and Rubin [11] is provided. Shin, Ahn and Senoy [18] dicusses concerns in both Microsoft Passport and Liberty Alliance in terms of the general security requirements: confidentiality, integrity and availablility. What technology and methods should be used to achieve these objectives are stated. For instance, TLS/SSL should be used to achieve confidentiality. Additionally, privacy concerns are raised for FIM in general. How can one ensure that only relevant information is shared across FIM?

3.2 OAuth

The Devil is in the (Implementation) Details: An Empirical Analysis of OAuth SSO Systems [21]: The Empircal Analysis of OAuth focuses on analysing the implementations of OAuth. A number of different practical attacks on OAuth are devised and then they are tested on a number of different real world implementations of OAuth. Because the implementations of OAuth deviate, the attacks are successful on some implementations and unsuccessful on others. The attacks tested include eavesdropping on connections, XSS, CSRF and session swapping.
Other OAuth Contributions: Chari, Charanjit and Roy use the Universally Composble framework to model OAuth [4] . The purpose of the model is to show that OAuth is secure if the implementation correctly follows the specification of OAuth. Refinements to the OAuth protocol are suggested through the outcome of the work. Suhas et al. performs formal analysis of OAuth using the Alloy Framework [15]. The work suspects that if SPs in the OAuth protocol do not securely store secret credentials then malicious applications can be used in the name of the SP. The model which is built verifies that the flaw exists but no attacks are discussed that could exploit the flaw.

3.3 OpenID

Robust defenses for cross-site request forgery [3]: This paper describes a new attack which is named as a login CSRF attack. This is where an adversary maliciously logs an honest user into the adversarie's account. OpenID is mentioned as a protocol which is vulnerable to this attack but this paper is not mainly about OpenID. One security incident is identified in OpenID and since OpenID is a paper we consider we document it here.
Systematically breaking and fixing openid security: Formal analysis, semi-automated empirical evaluation, and practical countermeasures [22]: OpenID was formalised using HLPSL and then validated using AVISPA. Concrete attacks were designed to exploit the vulnerabilities exposed through verification for real world implementations of OpenID. The attacks designed were found to be successful on a number of different SPs. The paper lists seven attacks in total but some are variations of the same attack so for concise results we listed one of the variations of each attack. The attacks that are not counted have the same vulnerabilities and attack class. There may be some small difference in how a vulerability is exploited, but the attack class is the same it is functionally the same security incident. Note that in incident 2, the unencrypted communications vulnerability is implicit because although it is not stated that unencrypted communications is used, unencrypted communications must be used to modify the message.
Security Analysis of OpenID [20]: Focuses on manipulating OpenID assertions which are not protected by SSL.
Other OpenID contributions: Delft and Oostdijk conduct a survey that deeply looks at the issues in OpenID that have been found by others [23].

3.4 Facebook Connect

Formal analysis of Facebook Connect single sign-on authentication protocol [12]: Facebook Connect is a proprietary protocol which is why it has been somewhat difficult to analyse. Regardless, Facebook Connect has been analysed based upon the messages observed in transit using HLPSL and AVISPA.
Other Facebook Connect contributions: Egelman discusses the privacy/convenience tradeoff concerning facebook connect [5].

3.5 SAML

Security analysis of the SAML single sign-on browser/artifact profile [8]: abstractly models the SAML protocol by showing the sequence of messages that are sent and received throughout the protocol. This sequence is then reasoned about in order to deduce a number of attacks that could be executed on the SAML protocol.
SAML Artifact Information Flow Revisted [9]: This paper is intended as a follow up to the paper commented on above. The follow up paper states that while the issues pointed out in the first paper have been addressed some issues still remain.
Formal Analysis of SAML 2.0 Web Browser Single Sign-On: breaking the SAML-based Single Sign-On for Google Apps [1]: The formal model checker SATMC was used to check the SAML implementation of Google single-sign on. A MITM attack was found where a dishonest SP can use access credentials from a legitimate user to login to a different SP.

3.6 Liberty Alliance

Analysis of liberty single-sign-on with enabled clients [16]: Abstractly models Liberty Alliance to find a Man-in-the-Middle (MITM) attack. Since Liberty is based on SAML, it is not surprising that this attack has also been found in SAML [1].
Other Contributions: There is work that consisted of research both in Microsoft Passport and Liberty Alliance [18] discussed in the Passport subsection of this section.

4 Breakdown of Security Incidents in FIM

In this section we show a breakdown of what problems have been found in the FIM protocols we consider. Refer to the technical report for in depth definitions for the different issues found [19].

4.1 Vulnerabilities

Refer to table 1 for an overview of vulnerabilities in FIM.

Table 1. Overview of vulnerabilities in FIM

	Passport	OAuth	OpenID	Facebook	SAML	Liberty
Inadequate browser defence	x					
User mistakes	x					
Unencrypted communications	x	x	x	x	x	
Weak DNS	x				x	
Centralised point of attack	x					
Vulnerable SP		x	x			
Automatic authorisation		x				
Lack of Authentication		x	x		x	x
Message formatting					x	

4.2 Attack Classes

Refer to table 2 for an overview of attack classes in FIM.

Table 2. Overview of attack classes in FIM

	Passport	OAuth	OpenID	Facebook	SAML	Liberty
Bogus merchant	x					
Message modification	x	x	x		x	
DNS poisoning	x				x	
DoS attack	x					
XSS		x				
Session swapping		x	x			
CSRF		x	x			
Replay attack			x	x	x	
MITM					x	x

4.3 Component Failures

Refer to table 3 for an overview of component failures in FIM.

4.4 CIA Failures

Refer to table 4 for an overview of CIA failures in FIM.

5 Observations from Survey

5.1 Component Failures - Who is at Fault?

There exists security incidents that can only occur when agents within a FIM system behave in a certain way. There are other security incidents that happen

Table 3. Overview of Component Failures in FIM

	Passport	OAuth	OpenID	Facebook	SAML	Liberty
Browser	x				x	
SP		x	x	x		
IdP	x	x	x			
Protocol		x	x		x	x
External Service	x				x	

Table 4. Overview of CIA Failures in FIM

	Passport	OAuth	OpenID	Facebook	SAML	Liberty
Confidentiality	x	x	x	x	x	x
Integrity			x			
Availability	x					

when all of the agents behave correctly, it is therefore the sole responsobility of the protocol as all the agents performed their function. The question is, when is it the responsobility for the protocol designers to make changes and when is it the responsobility for the agents to increase security measures?

We have identified when a security incident can be reasonably stopped my adding security measures on the relevant components. The exception to this is when an external service is involved, like the DNS, in this case neither the protocol or the components are immediately responsible and the problem is a deeper problem with modern infrastructure itself. When it is necessary for the adversary to have a certain level of privilege on a component, then it is usually the components fault. For instance, in a XSS attack the adversary needs to compromise the SP in some way. In the case of XSS it can then be said that the fault lies with the SP as they have failed to secure themselves from the XSS attack. However, when the adversary does not need explicit powers over components and attacks are still possible then the fault lies with the protocol as the components are behaving reasonably. Deciding on where the fault lies is a good use case of MAFTIA.

5.2 Causality Chains

From the survey we have conducted it is clear that some relationships exist between vulnerabilities, attack classes, component failures and CIA failures. We have observed that across FIM protocol boundaries the same attack classes have been seen to be possible on the same vulnerabilities. What this allows us to do as an observer is identify causality chains for security incidents in FIM. The information gleaned from such observation is useful in then allowing us to see which vulnerabilities need to be fixed to prevent certain attacks. In this subsection we present how causality chains are identified, what causality chains we have discovered and an overall picture of the progression of vulnerabilities to attacks to failures in FIM.

We identify causality chains when the same vulnerability, attack class, component failure and CIA failure are seen across research boundaries. By research boundaries we mean across different protocols or different authors. We have discovered a number of different causality chains that we will list below.

C1: We have seen vulnerable SPs being targetted by CSRF attacks in both OAuth and OpenID [21] [22]. The component that fails is the SP as it the component breached by the attack. The CIA failure that occurs is confidentiality as these attacks are used with the end goal of logging into user accounts.

C2: It is not always the case that an adversary logs into a user account to cause damage. We have seen examples of it being possible for a user to unknowingly sign into an adversary account when there is a lack of authentication vulnerability present. This is known as session swapping and can be seen in OAuth and OpenID [21] [22] . The protocol is at fault for not providing sufficient authentication and when the user is logged into an adversaries account they are in a precarious situation as their confidential details can be disclosed.

C3: It is not a surprise that two of the same security incidents have been found on SAML and Liberty as Liberty is based on SAML. Regardless, these protocols have been shown to be vulnerable to a MITM attack because of a lack of authentication provided by the security tokens in these protocols [1] [16]. The component that fails is the protocol itself as there is nothing any of the other components can do to prevent this attack. Confidentiality is breached in the user as an adversary can access user details from honest SPs.

C4: When there is nothing that binds a message to the sender, a lack of authentication is present and a replay attack is possible. This has been seen to happen in OpenID and SAML [22] [8]. The protocol component itself fails for not providing means for messages to be bound to their senders. The CIA property to fail is confidentiality as adversaries use the replay attack to login to accounts in the legitimate users name.

C5: An easy vulnerability to attack is unencrypted communications. One particular way that is prevelant is through message modification. Message modification can be used to present legitimate access tokens to a SP as an adversary which can be seen in OAuth and SAML [21] [8]. In OpenID we see how it can be used to change profile information in transit [22]. The component failue varies because it depends on the endpoint which is responsible for the end-to-end encryption. The CIA failure can also depend because if an adversary is able to modify a message in a way to present themselves as a legitimate user confidentiality is breached, if an adversary changes parameters it is an integrity issue.

5.3 Protecting FIM Systems

In 5.1 we indicate how using MAFTIA we can decide which component is at fault. This is useful, but there also needs to be a discussion of what can be changed to prevent successful attacks from occuring. In 5.2 we identify prevelant causality chains in security incidents. The authors who found those chains often

suggested relevant defences to prevent the security incident happened. We list corresponding defences for the causality chains identified.
CD1: This defence refers to CSRF attacks seen in OAuth and OpenID [21] [22]. Since both the OAuth and OpenID studies were practiced on real systems, it is surprising how many systems are vulnerable. However, it is the case that the protocol can be changed to stop these attacks being effective. The OpenID study states that a possible defence is to bind user requests to the session taking place. The SP hashes a secret generated by the SP together with the session id to create a token. This token is appended as a hidden field to a form rendered by the SP to ensure that any form submission does come from the expected sender.
CD2: This defence refers to session swapping attacks seen in OAuth and OpenID [21] [22]. In a similar vein as CD1, the OAuth study points out that the solution is to bind the session to the browser itself. Hash the session cookie at the start of the protocol run and each time the SP receives a new message the hash of the session id is checked again.
CD3: This defence refers to the MITM attacks seen in SAML and Liberty [1] [16]. The attack is possible because access tokens are not explicit to single SPs and can be used at other SPs. The simple solution is to bind tokens to the SP it is meant for as is described in the Liberty study. However, there are other measures suggested such as providing a list of safe SPs.
CD4: This defence refers to the replay attacks seen in OpenID and OAuth [22] [8]. Both studies cite the stateless nature of the connection as a problem and put forth the idea of considering the senders IP address before granting access. This solution does have its problems, as in some cases, different users would appear to be communicating from the same IP address if a web proxy is used.
CD5: This defence refers to the message modfication attacks seen on OAuth, OpenID and SAML [21] [22] [8]. The obvious solution to this attack is to force SSL communication end to end. It is difficult to offer general solutions across protocol boundaries as other proposed solutions are specific solutions for the protocol being considered. The defence for this attack, use encryption, is worrying as it is something protocol designers have no control over. Furthermore, it has been shown in the OAuth and OpenID studies how insecure SPs can be in practice. FIM protocol designers unfortunately need to consider protocol runs with insecure hosts.

5.4 Significant Threats to FIM

Bogus merchants have been stated to be a problem in FIM [11]. Why this is concerning is because there is still no effective defence against bogus merchants.There have been attempts, such as browser addons [24]to help safeguard users from visiting malicious websites. The problem is with this solution is that even if the addon is effective, the average user is not likely to install such an addon.

The unencrypted communications vulnerability has been widely exploited in the papers we have studied. The vulnerability can sometimes appear in subtle ways like in SAML where it could be exploited at a very specific point in the protocol[8]. However, othertimes the vulnerability is much more blatant as has been shown in OAuth where a surprising amount of providers sent access tokens in the clear [21]. FIM providers need to ensure their endpoints are properly protected.

The lack of authentication vulnerability is another vulnerability that has caused a lot of problems in FIM. What this vulnerability means is that an adversary can send a message claiming the identity of a user they are not. For example, in SAML it was shown that a replay attack was possible because the receiver of a message had no way of knowing who that message came from.

There are deeper issues with the web that FIM protocol designers have no control over. The possibility of the DNS being leveraged by an adversary to break FIM systems has been stated by the community [11] [8]. While there may be clever countermeasures FIM protocol designers can devise to deflect problems such as reliance on DNS, this problem is largely outside the sphere of influence of FIM protocol designers.

6 Conclusion

We have created a framework that allows for a consistent and comparable viewing of different security incidents in FIM. Whilst others have studied specific protocols in FIM to find problems, we have moved across protocol boundaries to see the larger picture of incidents in FIM. We have shown the mistakes that have been made in FIM systems for the purpose of steering FIM systems away from making the same mistakes again.

References

1. Alessandro Armando, Roberto Carbone, Luca Compagna, Jorge Cuellar, and Llanos Tobarra. Formal analysis of saml 2.0 web browser single sign-on: breaking the saml-based single sign-on for google apps. In *Proceedings of the 6th ACM workshop on Formal methods in security engineering*, pages 1–10. ACM, 2008.
2. Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, et al. *Fundamental concepts of dependability*. University of Newcastle upon Tyne, Computing Science, 2001.
3. Adam Barth, Collin Jackson, and John C Mitchell. Robust defenses for cross-site request forgery. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 75–88. ACM, 2008.
4. Suresh Chari, Charanjit S Jutla, and Arnab Roy. Universally composable security analysis of oauth v2. 0. *IACR Cryptology ePrint Archive*, 2011:526, 2011.
5. Serge Egelman. My profile is my password, verify me!: the privacy/convenience tradeoff of facebook connect. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2369–2378. ACM, 2013.
6. Tewfiq El Maliki and Jean-Marc Seigneur. A survey of user-centric identity management technologies. In *The International Conference on Emerging Security Information, Systems, and Technologies (SECUREWARE 2007)*, pages 12–17. IEEE, 2007.
7. Facebook. Facebook Connect Usage statistics, 2016.

8. Thomas Groß. Security analysis of the saml single sign-on browser/artifact profile. In *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*, pages 298–307. IEEE, 2003.
9. Thomas Groß and Birgit Pfitzmann. Saml artifact information flow revisited. In *In IEEE Workshop on Web Services Security (WSSS)*, pages 84–100. Citeseer, 2006.
10. Audun Jøsang, John Fabre, Brian Hay, James Dalziel, and Simon Pope. Trust requirements in identity management. In *Proceedings of the 2005 Australasian workshop on Grid computing and e-research-Volume 44*, pages 99–108. Australian Computer Society, Inc., 2005.
11. David P Kormann and Aviel D Rubin. Risks of the passport single signon protocol. *Computer networks*, 33(1):51–58, 2000.
12. Marino Miculan and Caterina Urban. Formal analysis of facebook connect single sign-on authentication protocol. In *SOFSEM*, volume 11, pages 22–28, 2011.
13. Daniel Olmedilla, Omer F Rana, Brian Matthews, and Wolfgang Nejdl. Security and trust issues in semantic grids. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2006.
14. Rolf Oppliger. Microsoft. net passport: A security analysis. *Computer*, 36(7):29–35, 2003.
15. Suhas Pai, Yash Sharma, Sunil Kumar, Radhika M Pai, and Sanjay Singh. Formal verification of oauth 2.0 using alloy framework. In *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*, pages 655–659. IEEE, 2011.
16. Birgit Pfitzmann and Michael Waidner. Analysis of liberty single-sign-on with enabled clients. *IEEE Internet Computing*, 7(6):38–44, 2003.
17. David Powell, Robert Stroud, et al. Conceptual model and architecture of maftia. *Technical Report Series-University of Newcastle Upon Tyne Computing Science*, 2003.
18. Dongwan Shin, Gail-Joon Ahn, and Prasad Shenoy. Ensuring information assurance in federated identity management. In *Performance, Computing, and Communications, 2004 IEEE International Conference on*, pages 821–826. IEEE, 2004.
19. Sean Simpson and Thomas Groß. Technical report of security analysis in fim protocols. *Technical Report Series-University of Newcastle Upon Tyne Computing Science*, 2016.
20. Pavol Sovis, Florian Kohlar, and Jörg Schwenk. Security analysis of openid. In *Sicherheit*, pages 329–340, 2010.
21. San-Tsai Sun and Konstantin Beznosov. The devil is in the (implementation) details: an empirical analysis of oauth sso systems. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 378–390. ACM, 2012.
22. San-Tsai Sun, Kirstie Hawkey, and Konstantin Beznosov. Systematically breaking and fixing openid security: Formal analysis, semi-automated empirical evaluation, and practical countermeasures. *Computers & Security*, 31(4):465–483, 2012.
23. Bart Van Delft and Martijn Oostdijk. A security analysis of openid. In *IFIP Working Conference on Policies and Research in Identity Management*, pages 73–84. Springer, 2010.
24. Min Wu, Robert C Miller, and Greg Little. Web wallet: preventing phishing attacks by revealing user intentions. In *Proceedings of the second symposium on Usable privacy and security*, pages 102–113. ACM, 2006.