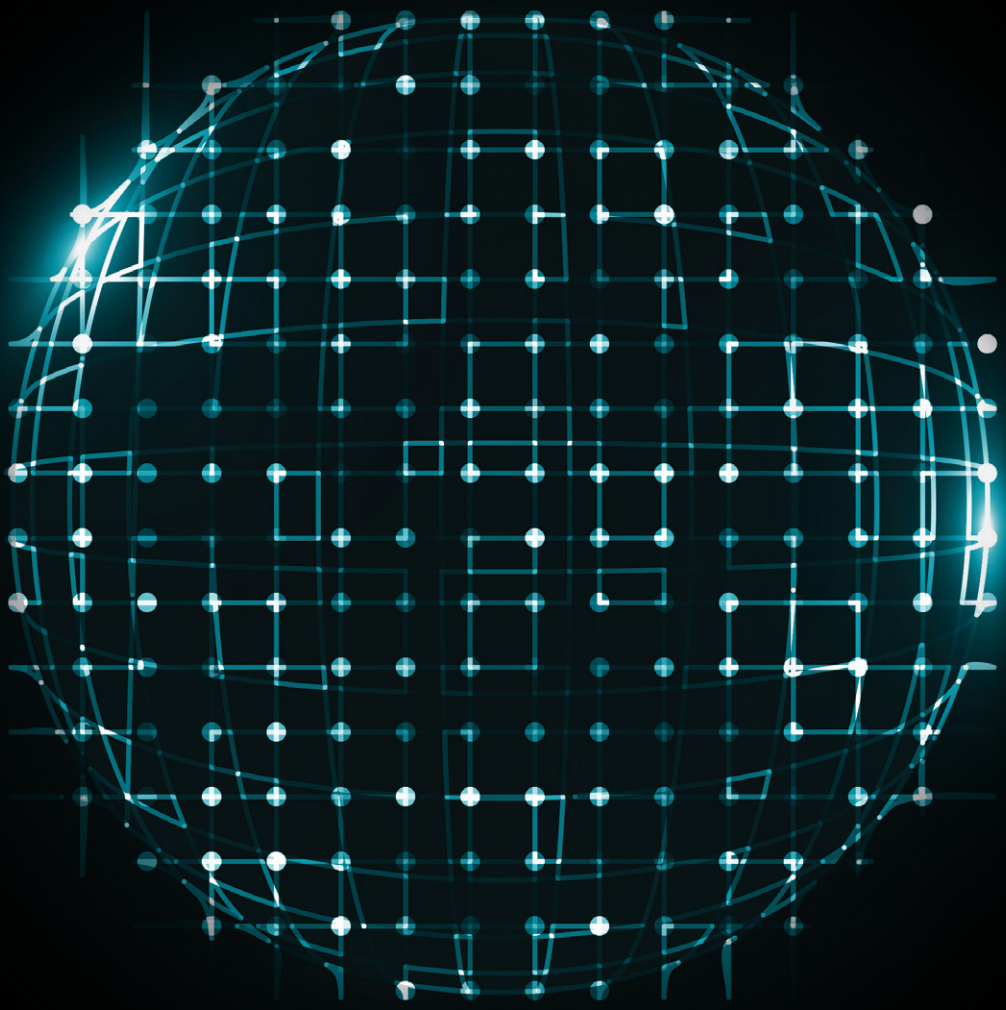
Deloitte.



SWIFT Systems and the SWIFT Customer Security Program

Crime has been present in the financial sector since its very beginnings. Today however, threat actors from all over the world can conduct any kind of attack (heist, espionage, sabotage) on any kind of bank.

Introduction

Cybercrime has seen a steady increase over the last decennia in the finance sector. However, it seems that attacks are increasingly targeting the financial messaging systems, such as SWIFT. Today the attacks on financial institutions are more sophisticated, advanced and executed with more delicacy, the gains are also much higher. This created a shift from putting focus on large groups of individual customers to larger individual institutions.

Countermeasures

SWIFT has introduced the Customer Security Program (CSP) as a countermeasure to these cybercrimes. However, it was also implemented to raise the bar of logical and physical security for the community as a whole. The Customer Security Control Framework (CSCF) of 2019 consists of a set of three objectives, which focus on seven principles and contain 31 controls. Customers must self-attest to these CSCF controls by 1 January 2020.

Based on our experience with the evaluation of the CSCF at several SWIFT customers, we will analyze SWIFT-related breaches and most common control failures in this document. Through this, we provide a set of recommendations on how to prepare for the self-attestation and how to secure your environment better.

SWIFT and what it does

Banks are connected to each other, creating a strong need for secure communication between them. To ensure standardized financial messaging exchanges in a secure way, SWIFT developed a messaging platform. Today, more than 11,000 customers in over 200 countries and territories are connected to the messaging platform, products and services of SWIFT. While reducing distance within the financial community on global, regional and local levels, SWIFT also defines standards to help shape market practices and to face issues of mutual concern.

Year	Number of FIN messages sent
2018	7,873,626,879
2017	7,076,457,019
2016	6,525,799,505

(SWIFT FIN Traffic & Figures 2019, 2019)

Evaluation of cybercrimes over the past years

The last couple of years has seen an increase in the number of cyber-attacks that we are aware of. There are more attacks that we are not aware of or that have been blocked because of effective controls. Here we summarize some of the most well-known and impactful attacks that happened over the last years, using the SWIFT system.

The Bank of Bangladesh

$951,000,000 stolen using trusted Windows Software.

In February 2016, the Bank of Bangladesh faced a major cyber attack resulting in $81,000,000 unrecovered. The attackers gained control over SWIFT systems by deploying trusted Windows software to the bank's internal systems. Potentially, $951,000,000 was at stake.

Far Eastern International Bank in Taiwan
$60 Million using tailored Malware.

In October 2017, the bank in Taiwan was the target of a bank heist involving almost $60 million. The hackers reportedly used tailored malware to generate SWIFT messages containing fraudulent information. Much of the stolen funds was recovered with help of its banking counterparts, however $500,000 remained untraceable.

Banco del Austro in Ecuador
$12,000,000 stolen with credential compromise.

In January 2015, attackers obtained the credentials of a bank employee and used these to access the employee's email account to alter SWIFT transfer requests. Transfers were made from several accounts of Wells Fargo, HSBC and Hang Seng Bank accounts totaling $12 million. In the course of the investigation however, $1.85 million dollars were recovered and returned.

Banco de Chile in Chili
$10,000,000 stolen using Buhtrap its MBR Killer malware

A destructive malware was released at the Banco de Chile in May 2018. It caused mayhem, distracting defenders from another attack on the crown jewels. Whilst all systems were down, fraudulent messages were sent for $10,000,000.

Globex State Bank in Russia
$940,000 at risk using system hacking

In December 2017, a Russian bank has spotted attacks targeting its SWIFT systems, the attackers being able to enter in their bank system. The hackers tried to steal 55 million rubles ($940,000), but were only able to steal $100,000 as the Russian bank detected the suspicious wire transfers.

Thien Pong Bank in Vietnam
$1,130,000 stealing attempt through PDF reader

In December 2015, attackers used malware in an attempt to steal $1,130,000 from the Thien Pong Bank. The malware affected the Foxit PDF reader and allowed it to modify statements, by converting PDF to XML. The bank's employees timely halted the attack due to the identification of suspicious SWIFT messages.

Numerous unnamed banks in Russia and Ukraine
Multiple attacks reported on Ukrainian and Russian Banks

In 2016, $10,000,000 was stolen from an unnamed bank in Ukraine. Currently, it what attack vectors caused the theft. However, multiple attacks are reported in Russia and Ukraine, leaving significant amounts of money stolen.

Most common requirement failures/misunderstandings

We will present a top-3 of most common control failures and misunderstandings. As elaborate as the CSCF is, there is a number of controls that could be interpreted incorrectly. Even when the implementation guidelines are very specific in what at least should be in place.

Conflicting duties on applications and systems

While evaluating users on applications, operating systems and networks, we came across some conflicts. Below we list some of the most common findings:

1. users without 4-eyes principles enabled on the messaging or communication interfaces;
2. combining functions of application administrator and operating system administrator; and
3. combining functions of operating system administrators and network administrators.

A compromise of credentials could have devastating results. Moreover, fraud can be easily committed without mitigating controls in place.

Incorrect configuration of multi-factor authentication

The requirement 4.2 reads that there should be multi-factor authentication for:

1. operating system administrators preferably on the secure zone boundary (jump server); and
2. on the individual SWIFT applications.

However, we often see that the multi-factor authentication is implemented on an earlier stage than the boundary of the secure zone or through a wrong second factor.

Incorrect scope of the secured zone

A secure zone is a zone on your network dedicated to the payment systems. Requirement 1.1 section b, specifically states how a secure zone should be set-up. However, it is not always evident. A secure zone is a dedicated zone protected by separate firewalls, which only includes necessary systems and software. Moreover, this zone should have boundary protection as specified in requirement 1.1 section c. More often than not, the secure zone is wrongly interpreted and includes unnecessary systems, software or simply does not exist.

Evaluation of attack methods

We can classify the attack examples above into different tactical approaches. We present below two of these attack strategies in more details. These two attack strategies are possibilities of what happened, however these are based on own experience and not on facts retrieved from before mentioned attacks, as those details are classified.

System hacking

The attacker will take control of the SWIFT system in order to issue any desired transaction.

Step 1: Access the bank system

In this scenario, we assume that the bank internal system is vulnerable, and that the attacker is able to access it. It allows him to install a custom software, e.g. a key logger.

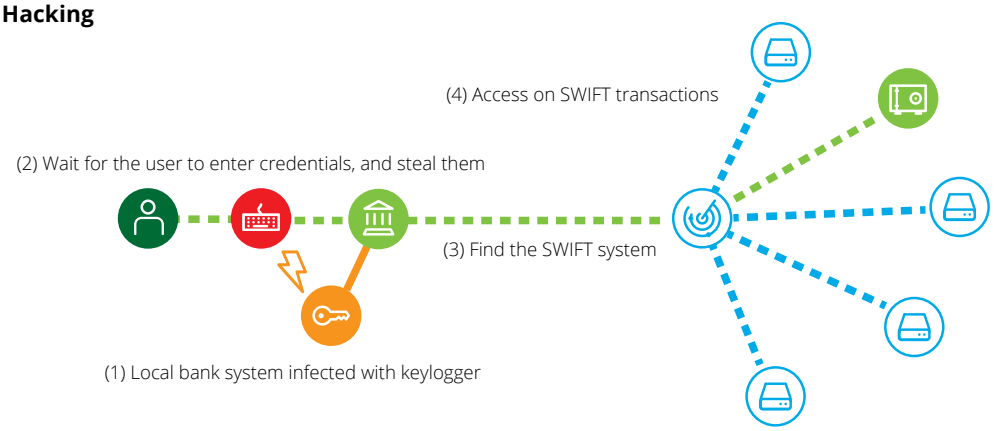
Step 2: Steal the credentials

Once an attacker has access to a system on which users can log in, he is able to steal user credentials with a key logger, a hidden software that will record all keyboard input from users. It will thus include usernames and passwords entered in the system.

Step 3: Lateral movement

The attacker has stolen the user accesses. We assume here that the accesses encompass a network access (e.g. the other machines/ systems connected to the bank internal system). The attacker last step is to scan the network in order to find the system he wants: the

Figure 1: System Hacking



SWIFT system. Once found, he can connect to it. He is now able to issue commands and transactions in the SWIFT system.

Tailored malware
The attack relies on a malware precisely customized for the software used by bank employees, modifying its behavior in a malicious way.

Step 1: Gather the information
First, the attacker must have the knowledge of the local system used by bank employees: which software is running (what is the version of the software, is it up to date?), for which purpose, etc. The attacker can acquire this knowledge through social engineering for example (i.e. asking an employee, spying, etc.)

Once this knowledge acquired, we make the fair assumption that the attacker will find some flaw in one of the programs found on the system.

Step 2: Infect the workstation
The software vulnerability found is considered interesting and the attacker will now exploit it. It is from then possible to modify the software behavior by adding additional instructions for the software inside the workstation, in the form of a malware.

Step 3: (Post) Exploitation
What the attacker could do now is to modify SWIFT transactions written by the targeted software. For example, changing the destinations in order to steal the money. An important aspect in this last step will be for the attacker to make sure not leaving any trace of his presence. Let us imagine that, on some case, it is not possible to change the transaction destination. Then, attackers would like to restore the original transaction and not leaving any error, in a way that the malware stays stealth on the system.

Evaluation of SWIFT cyber-attacks vs. OWASP 2017

The cyber threat landscape is growing faster nowadays, the attacks becoming more complex and elaborated every day. The Open Web Application Security Project (OWASP) established in 2017 the most common risks related to the cyber-attacks targeting web applications, and demonstrate that most of them fall into similar categories.

SWIFT is no exception to the cyber war, and undergoes several large-scale attacks every year. These are often performed with common modus operandi, including techniques such as tailored malware, compromise of credentials, email access and lateral movement.

These strategies encompass common attack vectors such as *SQL injections, Sensitive Data Exposure or Security Misconfiguration (see the Cyber Risks ranking involved in SWIFT Attacks below). These represent risks that each customer of SWIFT must take into account in its strategy.*

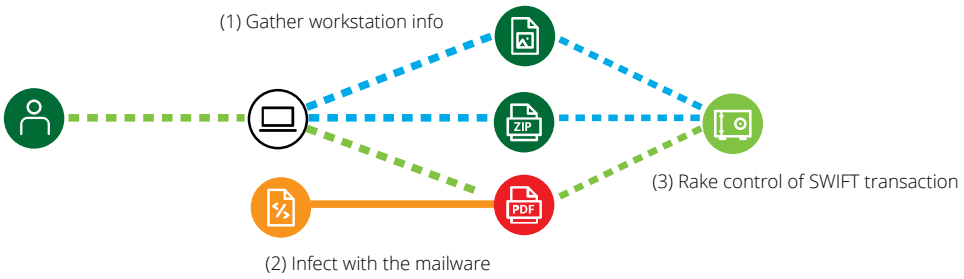
How can you prepare for the CSP self-assessment 2019

So how can a company prepare for the self-assessment of 2019.

1. Make sure you know your architecture type.
2. Know the difference between a protected zone and the SWIFT secure zone.
3. Make sure that all mandatory controls are in place, or you have a target date for implementation of those controls.

At Deloitte we have the expertise to provide you with input on the questions you might have. We can bring value to your company in several ways in the preparation phase, the assessment and in the mitigation phase.

Figure 2: Tailored Malware



SWIFT Cyber-Threat Landscape

Over the past years, significant cyber-attacks have been threatening the financial community and specifically, SWIFT. Both high- and low-profile cyber-attacks have resulted in significant financial loss.

Cyber-attacks come in different forms, however, are often caused by similar tactics. OWASP defined the top 10 most critical web application security risks for 2017. In the context SWIFT, attacks have been able to succeed due to weak links in the targeted financial institution's IT environment. This allows the attacker to make use of tactics such as SQL injections, malware deployment in insecure environments and altering email accounts through broken access controls.

The most impactful SWIFT cyber-attacks were described above, as well as the technical explanation of two of them.

OWASP top 10 security risks

1. Injection
2. Broken Authentication
3. Sensitive data exposure
4. XML External Entities (XXE)
5. Broken Access control
6. Security misconfigurations
7. Cross Site Scripting (XSS)
8. Insecure Deserialization
9. Using Components with known vulnerabilities
10. Insufficient logging and monitoring

(OWASP Top 10 Security Risks , 2018)

Contact us

Deloitte Luxembourg

Stéphane Hurtaud
Partner
Information &
Technology Risk
+352 45145 4434
shurtaud@deloitte.lu

Maxime Vérac
Director
Information &
Technology Risk
+352 45145 4258
mverac@deloitte.lu

Deloitte Global

Bert Truyman
Partner
SWIFT Directory of
the Assessors
+32 497 51 55 12
btruyman@deloitte.com

Deloitte.

46350