F-Secure

ANALYSIS: ATTACKING AND DEFENDING SWIFT SYSTEMS

SWIFT systems and the SWIFT Customer Security Program

An F-Secure Consulting guide
Author: Oliver Simmonet

CONTENTS

INTRODUCTION

The financial sector is and has always been a prime target for crime. In modern times, there is not only the risk of physical attacks, but also cyber attacks. Heist, espionage, and sabotage campaigns—once a threat which could be mitigated with the implementation of strong physical security controls and procedures—can now all be conducted by a wide range of threat actors, anywhere in the world.

Over the years, there has been a steady increase in cyber attacks on banks and the financial services sector as a whole, specifically with regards to the development and execution of advanced targeted attacks against financial messaging services, such as SWIFT. This comes as no surprise. Attackers have realized that focusing their resources on performing a low profile, calculated, and sophisticated attack on a financial institution has the potential for a much higher gain, requiring less overall effort than continuously targeting individual customers. As such, these attacks have increased over the years, not only in number, but also in sophistication; attackers are becoming increasingly persistent and adaptive when it comes to bypassing security controls and compromising critical financial systems to achieve their end goals.

Although a number of these attacks appear to be criminal in nature (e.g. the Carbanak gang), some attacks have shown strong links to nation states such as the Lazarus group (reportedly linked to North Korea). This may be an indication that large-scale financial heists are one of the few remaining methods of obtaining international currency within heavily sanctioned states.

As a countermeasure to the current cyber threat landscape, SWIFT implemented the Customer Security Programme (CSP). This requires all SWIFT customers to implement a number of controls defined by SWIFT's Customer Security Controls Framework (CSCF), to which customers had to first self-attest compliance before January 1, 2018. The framework has evolved and now outlines a growing collection of security controls to ensure that a minimal baseline for security is in place across all customers' local SWIFT deployments.

This report reviews a collection of major SWIFT-related breaches that have occurred over the years and analyzes the common factors shared between them. This is followed by an analysis of the scope and reliability of SWIFT CSP, identifying its strengths as well as its limitations. Finally, we provide recommendations on how to further secure these types of critical payment systems against future attacks.

WHAT IS SWIFT?

SWIFT (the Society for Worldwide Interbank Financial Telecommunications) is a secure messaging service used to transmit financial messages between member institutions around the world. SWIFT functions as a member-only cooperative service that is used and trusted by more than 11,000 financial institutions in more than 200 countries and territories around the world.

At its core, SWIFT provides access to the SWIFT messaging network (SWIFTNet) and its 4 messaging services (FIN, InterAct, FileAct and Browse). It also provides the standard for financial messaging and a range of solutions for the security, creation, management, processing, and validation of these messages.

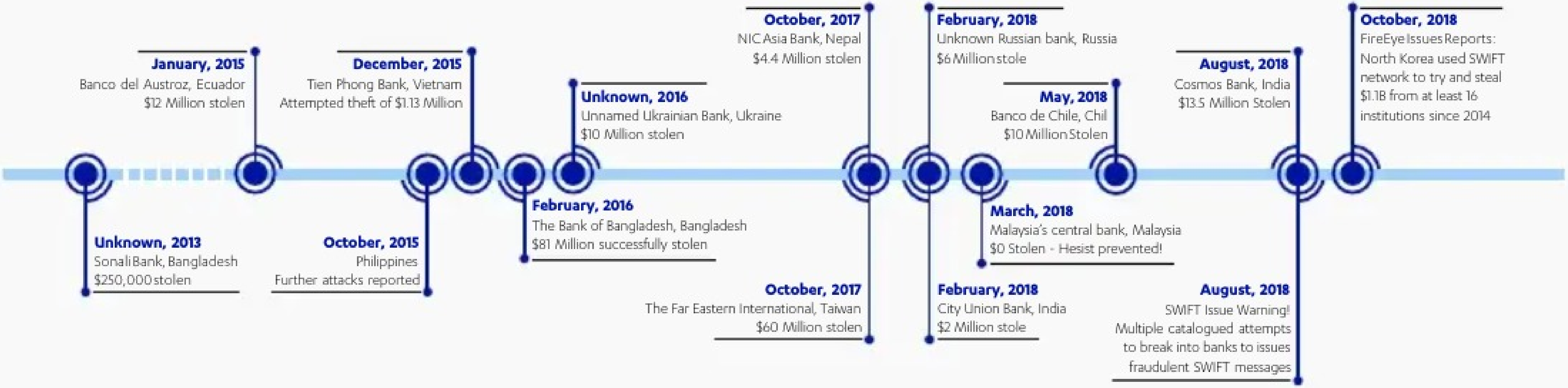
SWIFT does not, however, hold responsibility for the security of its customers' local SWIFT infrastructure, although it does provide assistance to ensure customers can manage cyber attacks. An example of this is the Customer Security Programme (CSP), which was originally introduced in late 2016.

Fig. 1. SWIFT logo

ATTACKS ON SWIFT SYSTEMS

There have been a significant number of high-profile attacks on SWIFT systems since 2013, with the majority of activity being observed between 2016 and 2018. Almost all these attacks resulted in significant financial loss.

Note: This section will focus on the 2013-2018 timeframe, when the most prominent attacks were disclosed. The number of successful attacks decreased significantly after 2018, likely due to the global success of the CSP programme.



Unknown, 2013
Sonali Bank, Bangladesh
$250,000 stolen

January, 2015
Banco del Austroz, Ecuador
$12 Million stolen

October, 2015
Philippines
Further attacks reported

December, 2015
Tien Phong Bank, Vietnam
Attempted theft of $1.13 Million

February, 2016
The Bank of Bangladesh, Bangladesh
$81 Million successfully stolen

Unknown, 2016
Unnamed Ukrainian Bank, Ukraine
$10 Million stolen

October, 2017
NIC Asia Bank, Nepal
$4.4 Million stolen

October, 2017
The Far Eastern International, Taiwan
$60 Million stolen

February, 2018
Unknown Russian bank, Russia
$6 Million stole

February, 2018
City Union Bank, India
$2 Million stole

March, 2018
Malaysia's central bank, Malaysia
$0 Stolen - Hesist prevented!

May, 2018
Banco de Chile, Chil
$10 Million Stolen

August, 2018
Cosmos Bank, India
$13.5 Million Stolen

August, 2018
SWIFT Issue Warning!
Multiple catalogued attempts to break into banks to issues fraudulent SWIFT messages

October, 2018
FireEye Issues Reports: North Korea used SWIFT network to try and steal $1.1B from at least 16 institutions since 2014

Fig. 2. Timeline: High-profile SWIFT-related attacks

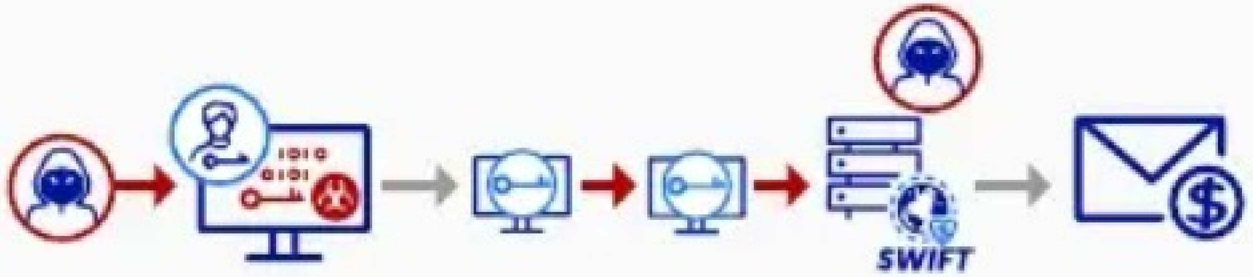
SONALI BANK
$250,000

Not much was known regarding the Sonali Bank heist (2013), and until 2016 it was treated as a 'cold case'. However, investigators re-opened the case after the attack on the Bank of Bangladesh in 2016.

It was reported that attackers were able to infect the bank's internal systems with key-logger software that was used to harvest user credentials. These credentials were then used to move laterally through the bank's network and gain access to the bank's internal SWIFT systems, where $250,000 worth of SWIFT transactions were made.

BANCO DEL AUSTROZ
$12,000,000

During the attack on Banco del Austroz (January 2015), attackers stole the credentials of an unnamed bank employee and used them to access the employee's Outlook email account. Using this access, the attackers located cancelled and rejected SWIFT transfer requests, altered their details, and reissued them, resulting in $12,000,000 worth of legitimate transfer requests being sent.

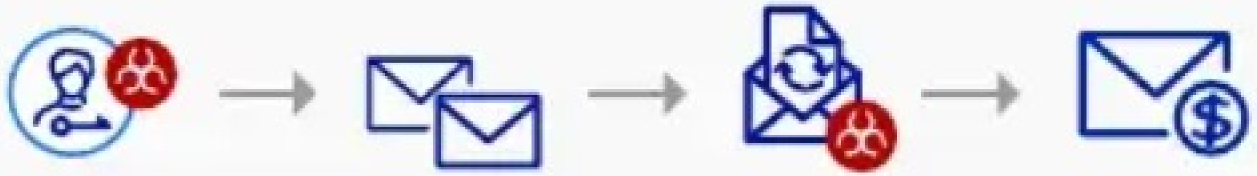


Attackers deploy a keylogger to steal employee credentials.

Using stolen credentials, the attackers move laterally across the network to find SWIFT-connected systems.

The attackers then issue an unspecified number of SWIFT transactions.

Fig. 7. Attack Path: Sonali Bank heist



Attacker steals employee credentials.

Attacker accesses employee outbox in search of rejected and cancelled transfer requests.

Attacker alters transfer request details (e.g., amount, destination).

Attacker re-issues transfer request messages.

Fig. 8. Attack Path: Banco del Austroz

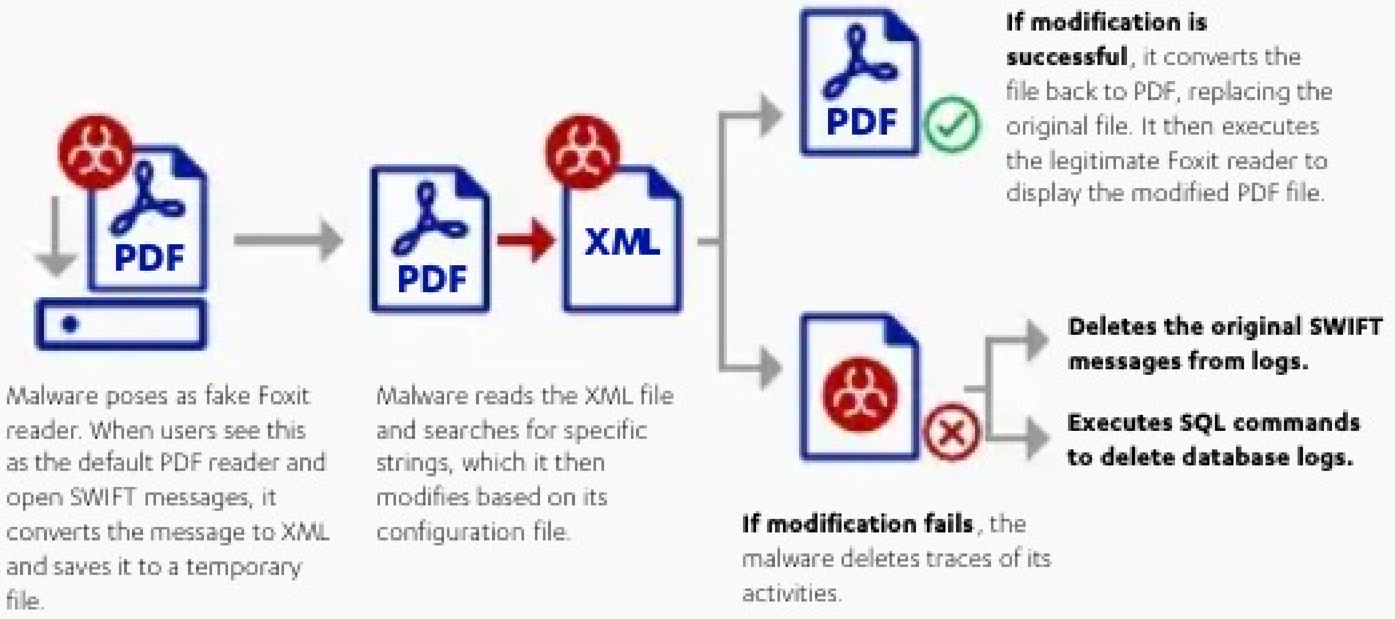
REPORTS FROM THE PHILIPPINES $UNKNOWN

In 2016, reports emerged that a bank in the Philippines had been the victim of an attack in October 2015. Although this attack occurred 2 months prior to the failed attack on the Tien Phone Bank in Vietnam (December 2015) and the attack on the Bank of Bangladesh (February 2016), malware samples recovered from all 3 incidents were linked. Furthermore, these malware samples were found to share similar code with malware used by the APT group Lazarus.

TIEN PHONG BANK $1,130,000 (ATTEMPTED)

During the attack on the Tien Phong Bank (December 2015), attackers used malware that specifically targeted the Foxit PDF reader, which was known to be used by the bank employees when viewing SWIFT statements. Attackers were able to install a malicious version of the Foxit PDF reader on employee workstations, which altered statements (when opened) in order to hide evidence of any malicious activity.

This malware was found to be installed on infrastructure provided by a third-party vendor, specifically used to provide the bank's connection into the SWIFT messaging network. Through a carefully planned and sophisticated attack, employees at the Tien Phong Bank identified suspicious SWIFT messages and rapidly contacted all parties involved. This prevented the transfer requests from being completed and the attempt to steal $1,130,000 was halted.



Malware poses as fake Foxit reader. When users see this as the default PDF reader and open SWIFT messages, it converts the message to XML and saves it to a temporary file.

Malware reads the XML file and searches for specific strings, which it then modifies based on its configuration file.

If modification is successful, it converts the file back to PDF, replacing the original file. It then executes the legitimate Foxit reader to display the modified PDF file.

If modification fails, the malware deletes traces of its activities.

Deletes the original SWIFT messages from logs.

Executes SQL commands to delete database logs.

Fig. 9. Execution of malware used in Vietnam hack

UNNAMED UKRAINIAN BANK $10,000,000

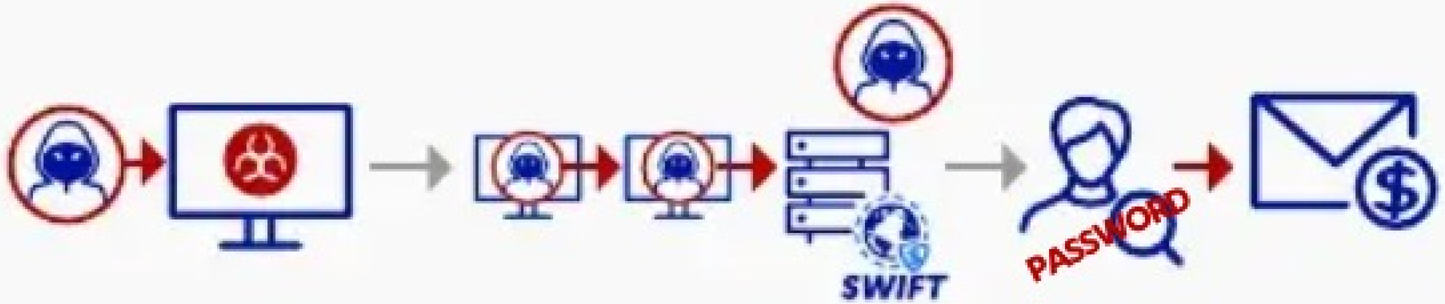
Details regarding the compromise of an unnamed Ukrainian bank (2016) are limited, though it was reported that $10,000,000 was stolen and that the attack was similar to that of the Bank of Bangladesh. It was further reported that this attack was only one of many that the Ukraine and Russia had experienced, resulting in the loss of "hundreds of millions of dollars".

THE FAR EASTERN INTERNATIONAL BANK $160,000 (OF $60,100,000)

In October 2017, an attack was carried out against the Far Eastern International Bank. During the heist, attackers used malware similar to that used by the APT group Lazarus, which, as reported, has been linked to multiple attacks on financial institutions around the world.

This malware was used to gain access to and move through the bank's internal network in order to infiltrate SWIFT systems. The attackers then compromised employee credentials and used this information to authenticate to the SWIFT Alliance Messaging Hub and issue a total of $60,100,000 worth of fraudulent transactions. Although it was initially understood that all but $500,000 was lost, the Financial Supervisory Commission (FSC) reported that the final amount lost by Far Eastern Bank was $160,000.

Following an investigation, it was found that the bank's security posture was not in line with the requirements outlined by Taiwan's banking law. As a result, Taiwan's financial regulator fined the Far Eastern International Bank $266,524, raising the total financial loss of the incident to $426,524.



Attackers deploy malware to access the bank internal network.

Laterally moves across the network in search of SWIFT-connected

Attackers compromise SWIFT operator credentials and use SWIFT

Fig. 10. Attack Path: The Far Eastern International Bank

THE NIC ASIA BANK
$580,000 (OF $4,400,000)

In October 2017, there was another attack on the NIC Asia Bank. Attackers specifically targeted the bank during the Hindu festival Tihar, one of Nepal's largest holidays.

According to reports, $4,400,000 of fraudulent SWIFT transactions were issued during the attack. However, NIC identified the suspicious activity and informed Nepal Rastra Bank (Nepal's central bank), resulting in the recovery of all but $580,000 of the $4,400,000.

At the time of writing, investigations into this attack are still ongoing. KPMG India's forensic team, who were commissioned by NIC Asia Bank to perform a digital investigation, had requested two additional weeks to complete their investigations.

CITY UNION BANK
$1,000,000 (OF $1,872,150)

An attack targeting SWIFT systems was reported in February 2018, with nearly $2,000,000 stolen from the City Union Bank in India. The attack had similar patterns to that of the Bang of Bangladesh in 2016, in that it first disabled the Bank's printer, preventing the bank from receiving acknowledgement messages. While the printer was disabled, the attackers issued 3 payment messages:

1. $500,000 to Dubai bank, via a New York Standard Chartered Bank. This, however, was blocked immediately.
2. A second message for $372,150 to Turkey, via a Standard Chartered account in Frankfurt. The Turkish lender involved blocked the transfer from being finalized.
3. A third payment for $1,000,000 was transferred via a Bank of America account (in New York) to a bank in China.

REPORTS FROM RUSSIA
$6,000,000

Also in February 2018, reports emerged from Russia's central bank stating that attackers had stolen $6,000,000 via SWIFT. This was stated to have been achieved by the attacker obtaining access to a computer within an unnamed Russian bank and using this access to transfer the money into their own account.

MALAYSIA CENTRAL BANK
$UNKNOWN (ATTEMPTED)

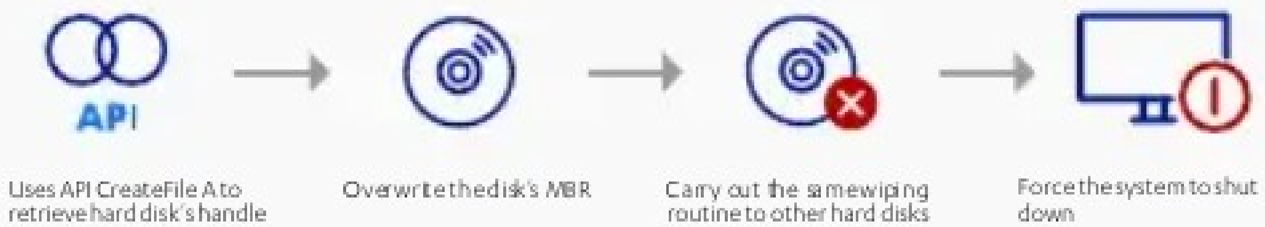
During March 2018, Malaysia's Central Bank disclosed that it had been the victim of a cyber attack, in which threat actors attempted to steal funds via fraudulent SWIFT payment messages. Although little information was released regarding this attack, Malaysia Central bank did state that: "All unauthorised transactions were stopped through prompt action in strong collaboration with SWIFT, other central banks and financial institutions".

BANCO DE CHILE $10,000,000

During an attack on Banco de Chile in May 2018, attackers disrupted a large number of bank systems. Initially, the attack did not seem related to any attempted heist and it was assumed that this initial wave of activity was simply destructive.

On May 24, hundreds of workstations and servers within Banco de Chile ceased functioning. During the response to the incident, it was found that MBR Killer had been deployed widely across their estate. This was a piece of malware that disrupted the Master Boot Record (MBR) of a computer's hard drive. Disruption to this section of a hard drive is catastrophic, as it is the first sector of the drive which is called before loading the operating system. Once disrupted, the computer is unable to boot.

Trend Micro, who analyzed the malware, outlined that it carried out the following 4 steps during its MRB-wiping routing (see diagram).



Uses API CreateFile A to retrieve hard disk's handle

Overwrite the disk's MBR

Carry out the same wiping routine to other hard disks

Force the system to shut down

Fig. 11. MBR Killer wiping routin

The affected systems were disconnected, affecting operations necessary to prevent the malware from spreading. However, during this period, a number of anomalous transactions were identified within their SWIFT systems.
At this point, the bank realized that the vast destruction to their systems was not the attack, but simply a distraction from the attacker's end goal of issuing fraudulent transactions. Some were prevented, but not all. Through utilizing fraudulent SWIFT transactions, the attack cost the bank $10,000,000.

COSMOS BANK $13,500,000

In August 2018, reports emerged that Cosmos Bank in India had fallen victim to a large scale, coordinated attack in which $13,500,000 was stolen via a combination of ATM withdrawals and fraudulent SWIFT payment instructions.

The malware deployed by the attacker targeted the bank's automated teller machine (ATM) server, resulting in 850 million rupees being extracted from ATMs across 28 countries. This was composed of 14,049 transactions. During the attack, an additional 129 million rupees were transferred to a company's account in Hong Kong via 3 fraudulent SWIFT payment messages.

FURTHER REPORTS AND NOTIFICATIONS

In August 2018, SWIFT issued a warning to its customers, stating:

"Swift is aware of a number of recent cyber incidents in which malicious insiders or external attackers have managed to submit Swift messages from financial institutions' back offices, PCs or workstations connected to their local interface to the Swift network".

Following this, in October, FireEye released a detailed report on APT38. Among other things, this outlined that North Korea had been using the SWIFT network in various attempts to steal around $1.1B from at least 16 institutions since 2014.

COMMON FACTORS

In review of these high-profile attacks, the first common factor is that almost all of them involve the deployment of some type of malware onto a bank's internal systems. Furthermore, we see that attackers frequently pair this with the compromise of user credentials. A list of the key tactics used are as follows:

- Phishing
- Malware deployment
- Keylogging
- Custom developed tooling
- Study of the environment and security processes
- Credential compromise
- lateral movement
- Email access
- Possible insider threat
- Abuse of business processes

Overall, none of the attacks directly compromised the SWIFT network itself, and were instead the result of flaws within the security controls deployed across the targeted bank's IT environments. Furthermore, attacks were frequently paired with some type of user error. This comes as no surprise, however. A widely accepted statistic highlights that that 95% of all incidents recognize "human error" as a contributing factor. This is due to the fact that, regardless of how strong the security of a system is, many security controls can be bypassed by human error. As an example, a password used to access a secure system or resource, no matter how complex, will only remain secure if kept secret. If the password is inadvertently or deliberately disclosed, this will undermine any authorization and authentication controls implemented to restrict access to that system or resource.

It should be noted however, that although the above high-profile case studies all follow this particular pattern of well organized and involved attacks, these are only the few prominent and successful attacks which reach the media. There are many-many other attacks which happen each year targeting financial institutions. The risk, in regard to SWIFT systems, highlights that the focus should not always be the protection of payment operators who can issue payment messages, but also users from the broader environment. Attacks can be just as impactful by abusing lower-level components such as message queues to introduce new payments, or even front-end application-level vulnerabilities such as cross-site request forgery.

The defense against these types of threat actors must begin with a strong security model that is deployed and maintained across an organization's entire estate.

WHAT IS SWIFT CSP?

As a countermeasure within the current cyber threat landscape, SWIFT introduced the Customer Security Programme (CSP) to support SWIFT customers in securing their local SWIFT infrastructure. It requires that they implement a set of mandatory and advisory security controls outlined within SWIFT's Customer Security Controls Framework (CSCF). These controls have been identified by SWIFT based on cyber threat intelligence and in collaboration with industry experts, and are articulated around three main objectives:

1. Secure Your Environment
2. Know and limit access
3. Detect and respond

The CSCF initially outlined 27 control in 2017, 17 of which were mandatory. As the years have progressed, so has the framework. Version CSCFv2022 outlines 32 security controls (23 of which are mandatory), with each mitigating one of the specific risks that SWIFT customers face:

1. The unauthorized sending or modification of financial transactions
2. The processing of altered or unauthorized SWIFT inbound transactions
3. Business conducted with an unauthorized counterpart
4. Breaches in confidentiality (business data, computer systems, or operator details)
5. Breaches in integrity (business data, computer systems, or operator details)

Collectively, these 32 controls create a "Secure Zone" in which, at a minimum, all local SWIFT infrastructure resides. This isolates all local SWIFT systems from the wider enterprise network and places an emphasis on the security of all systems within it. The controls used to establish the zone are grouped across 8 specific principles:

1. Restrict internal access and segregate critical systems from the general IT environment
2. Reduce the attack surface and vulnerabilities
3. Physically secure your environment
4. Prevent compromise of credentials
5. Manage identities and segregate privileges
6. Detect anomalous activity to systems or transaction records
7. Plan for incident response and information sharing

The application of these security controls varies based on the SWIFT infrastructure located locally within an institution's environment. In recognition of this, SWIFT has grouped architectures into 4 main models, with some variation:

- **A1.** Formerly known as a "Full Stack" architecture. Both the messaging interface and communication interface are within the customer's local environment.
- **A2.** Formerly known as a "Partial Stack" architecture. The messaging interface is within the customer's local environmen, but a service provider manages the communication interface.
- **A3.** SWIFT Connector Architecture. Only a SWIFT software component (e.g. Alliance Lite2) is present within the local infrastructure, which is used to connect to a SWIFT service provider.
- **A4.** Customer Connector Architecture. This defines a "Connector" architecture that utilizes a non-SWIFT provided connector solution, such as IBM MQ.
- **B1.** No Local User Footprint Architecture. No SWIFT-specific infrastructure components are hosted within the customer's local environment.

The various components in scope of the CSCF are broken down as follows:

Secure Zone: a segmented portion of the network isolating SWIFT systems from the rest of the enterprise environment.

Messaging interface: a software product (e.g., Alliance Access) supporting the use of SWIFT's messaging services. This is typically connected directly to the Communication Interface.

Communication interface: a software product (e.g., Alliance Gateway) that provides a link between the SWIFT network (SWIFTNet) and the messaging interface software.

Connector: a local software product (e.g., Alliance Lite2 AutoClient) that facilitates communication with a messaging and/or communication interface.

SWIFTNet Link (SNL): a mandatory software product for accessing messaging services over a secure IP network (within fig. 12., the SNL is part of the communication interface).

HSM & PIK: the SWIFT Hardware Security Module and Public Key Infrastructure.

Network Devices: firewalls, routers, etc., within or surrounding the SWIFT Infrastructure.

Graphical user interface (GUI): software that produces the graphical interface for a user (e.g., Alliance Web Platform).

The Relationship Management Application (RMA): a SWIFT-mandated filter that enables customers to define which counterparties are permitted to send FIN messages to the institution.

Operators: individual end users and administrators who directly interact with the local SWIFT infrastructure.

Operator PCs: the end users' or administrators' computer devices, used to operate or maintain the local SWIFT infrastructure.

Data Exchange Layer: the flow of data between the upstream systems/middleware and the local SWIFT infrastructure.

Middleware Server: local middleware system implementations, used for data exchange between the SWIFT-related components (in the local SWIFT infrastructure or at a service provider) and the user back office.

The components which are not in scope of the CSCF are:

Back Office Systems: the systems responsible for business logic, transaction generation, and other activities that occur before transmission of data into the local SWIFT infrastructure.

General Enterprise IT Environment: the general IT infrastructure used to support the broad organisation (e.g. Mail server, directory services, employee PCs, etc.)

If implemented to its fullest extent, the CSCF should effectively isolate all local SWIFT infrastructure from the wider enterprise environment, leaving only the communication channel from the upstream systems and the middleware as an entry point.

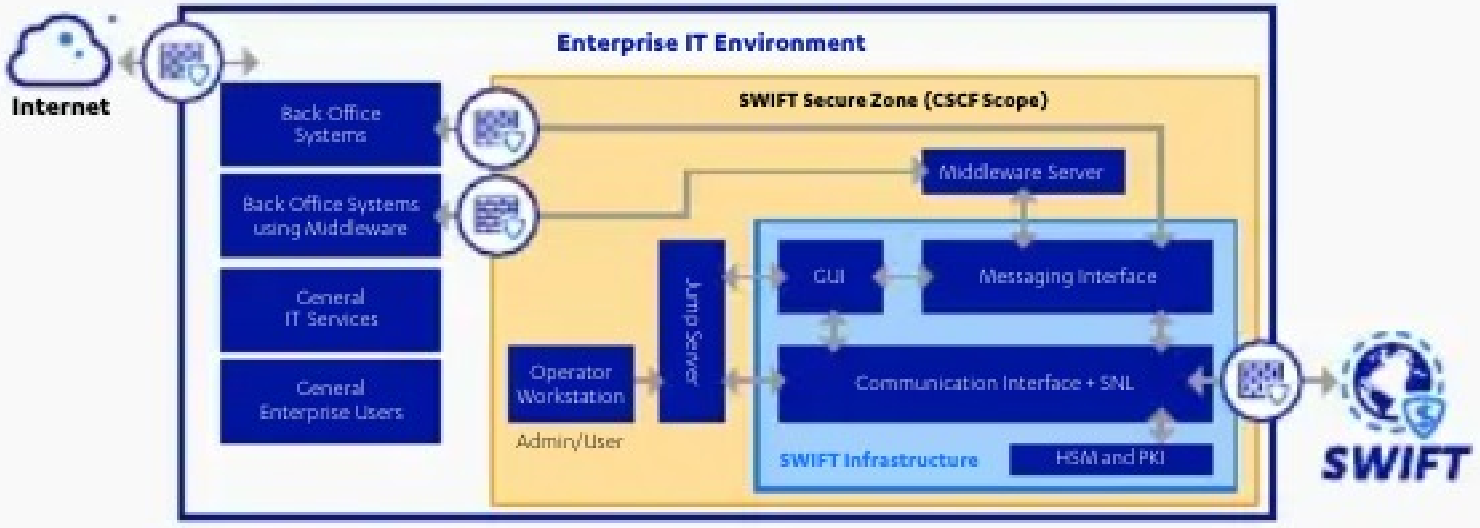
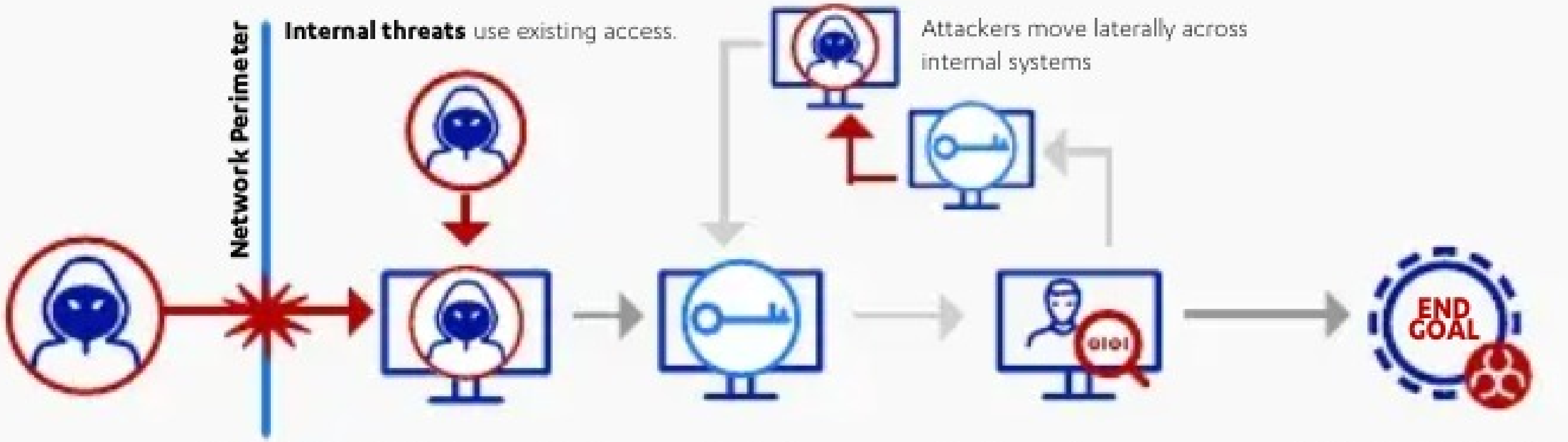


Fig. 12. A1 SWIFT Infrastructure

IS SWIFT CSP COMPLIANCE ENOUGH?

Implementing all mandatory (and advisory) controls specified by the CSCF has demonstrated that it can will greatly improve the security of local SWIFT infrastructure deployments and also ensure that all SWIFT customers have a baseline standard for their security. However, it should not be seen as a perfect solution to preventing all attacks. **Within the CSP document itself, it is stated that CSP should not be considered an exhaustive approach to security and it does not replace a well-structured security and risk framework.** By design, its purpose is to provide a baseline standard for the security of all local SWIFT systems, only. As such, general attack methodologies can still be applied to the most secure of critical systems. A "baseline" applies universally as a fundamental basic standard and it can't cater for the specifics of every individual environment. In the case of CSP, this baseline only covers SWIFT systems and can't provide overall protection across an entity's estate.

SWIFT systems are, and will remain, high profile targets for all threat actors operating with financial motivations. Regardless of the implementation of standard or advanced security controls, there is still a significant risk that these systems will have flaws that can be identified, targeted, and exploited by advanced and persistent threat actors. This is simply a consequence of the very complex nature of infrastructure deployed within financial institutions.



Internal threats use existing access.

Attackers move laterally across internal systems

External threats breach network perimeter and gain access to internal systems.

Attackers escalate their current user privileges.

Attackers perform reconnaissance activities to identify targets.

Attackers execute actions on objectives.

Fig. 13. General attack methodology

The main focus of the CSP is to isolate all SWIFT systems into the Secure Zone. Without this type of security model, attackers could have the opportunity to access SWIFT systems from any number of locations within the general enterprise network. The attack surface of this type of environment is represented in fig. 14.

With the implementation of all controls within CSCF, the attack surface of the SWIFT infrastructure is considerably reduced, removing a number of attack paths that could previously be exploited to access key systems. However, these controls do not render SWIFT systems impenetrable; a connection from the local SWIFT infrastructure (Secure Zone) to the financial institutions back-office systems must still remain. This is the weakest link in the security of all local SWIFT deployments.

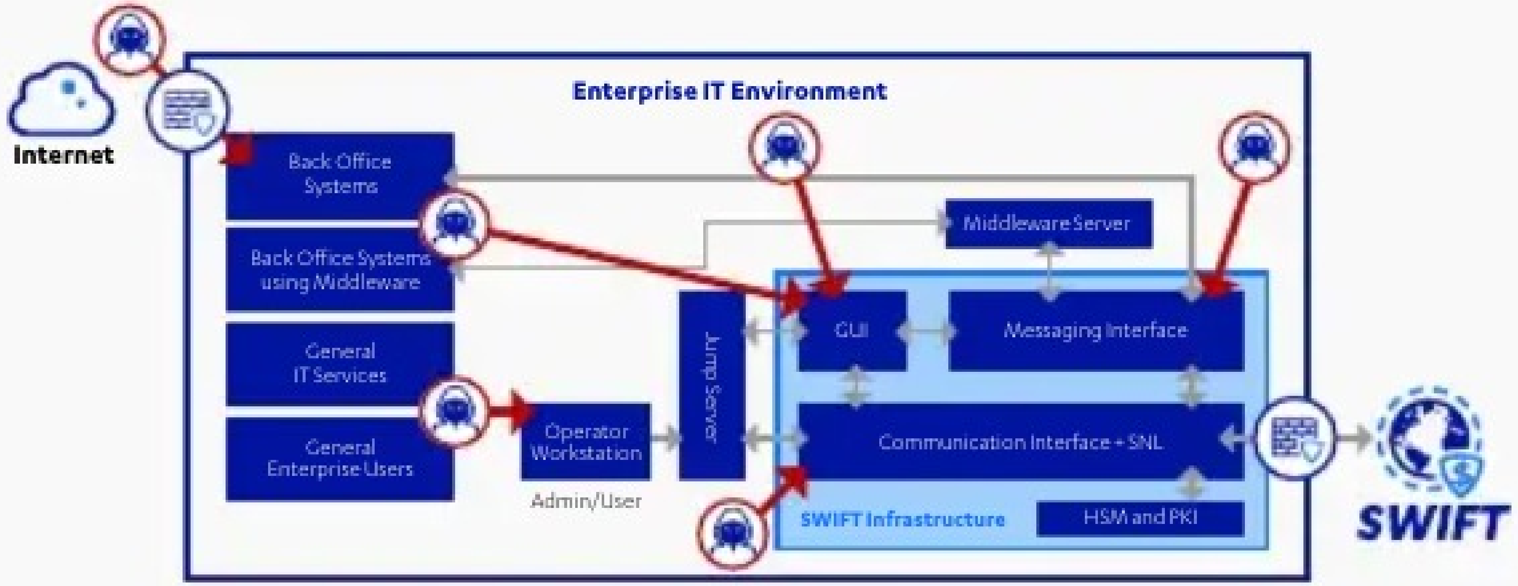


Fig. 14. Attack Vector: No Secure Zone

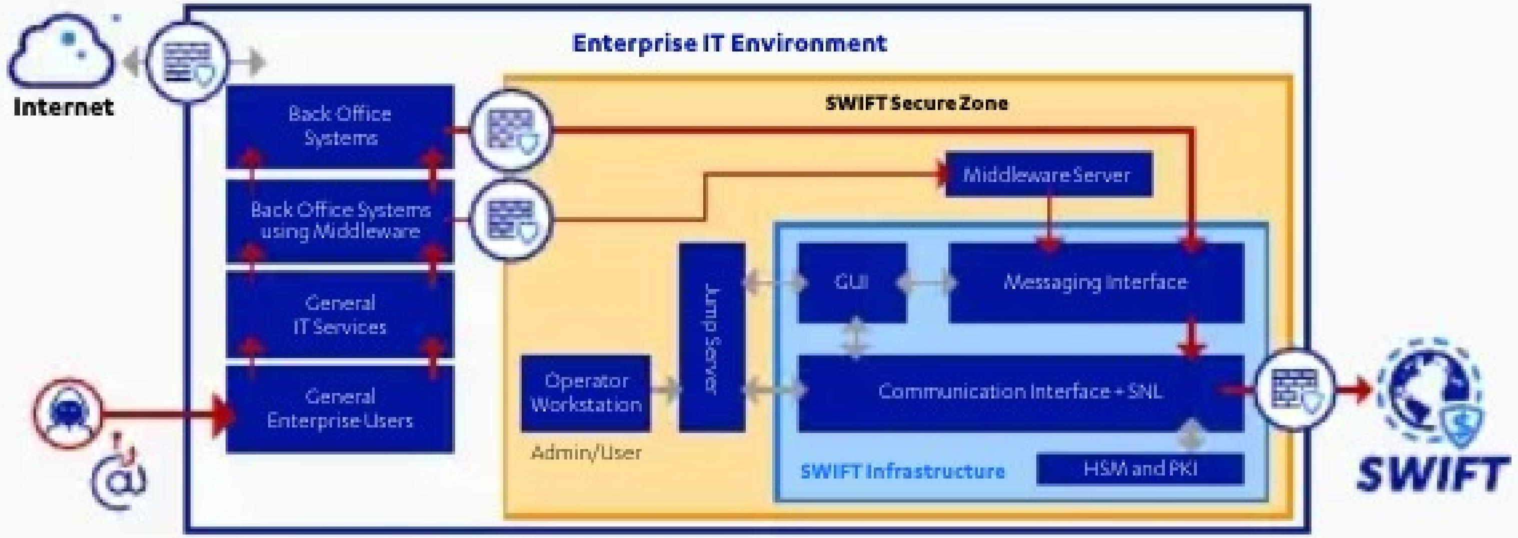
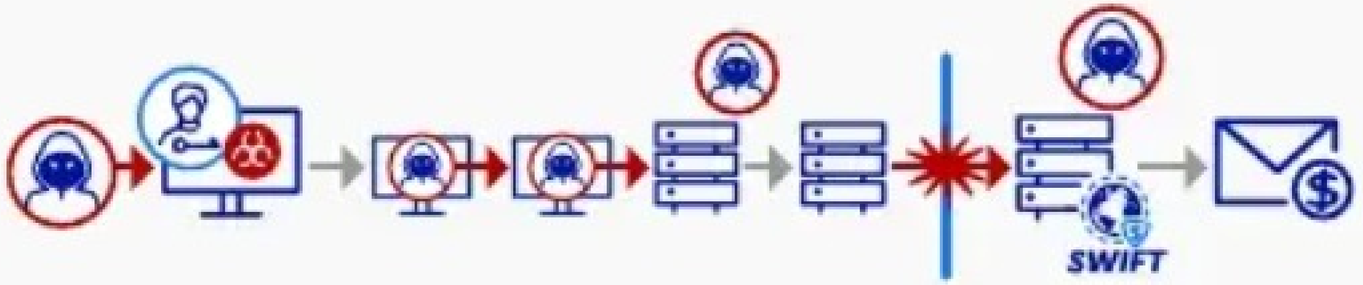


Fig. 15. Attack Vector: SWIFT CSP

The potential attack described on the previous page is broken down and generalized in the following diagram:



Attackers compromise bank network via vulnerabilities within external facing systems or via existing access.

Attackers move laterally across network in search of upstream SWIFT systems.

Attackers exploit upstream systems to breach the perimeter of the SWIFT Secure Zone, gaining access to the bank's local SWIFT infrastructure.

Attackers then exploit SWIFT users and/or applications to issue fraudulent transactions.

Fig. 16. Hypothetical attack path of a SWIFT infrastructure compromise

Analysis of this attack shows that the methodology still applies even with all CSP security controls in place. A mapping of the attack to the methodology is as follows:

1. Compromise the network perimeter and establish a foothold within the local network
2. Escalate current privileges (e.g. via system exploits or by obtaining user credentials)
3. Perform reconnaissance activities to identify the next target system
4. Repeat to move laterally across the network in search of the end goal (SWIFT upstream systems)

The process is then repeated to further compromise the isolated SWIFT systems:

1. The attackers breach the SWIFT network perimeter and establish a foothold within the network, then:
2. Escalate privileges in order to gain access to SWIFT system functionality
3. Perform reconnaissance to understand how transactions can be performed and authorized;
4. Execute their end goal (submission of fraudulent transactions).

Although a high-level overview of such an attack, this methodology of gaining direct access to SWIFT systems is not the only viable route for attacks. In 2020, F-Secure published research demonstrating the viability of creating fraudulent SWIFT messages without directly accessing any systems within the Secure Zone. This was achieved by abusing the underlying Message Queue (MQ) technologies used to transport messages to the Secure Zone. In the event an attacker gains the privileges of a single MQ administrator within the wider corporate network, they may have enough access to write forged SWIFT MT messages directly onto the queues. A diagram outlining the attack can be seen on the right in fig. 17.

The research emphasized the importance of ensuring that the overall standard of an organization's security is high enough to protect your highest privileged users from being compromised, even if they operate outside of the Secure Zone.

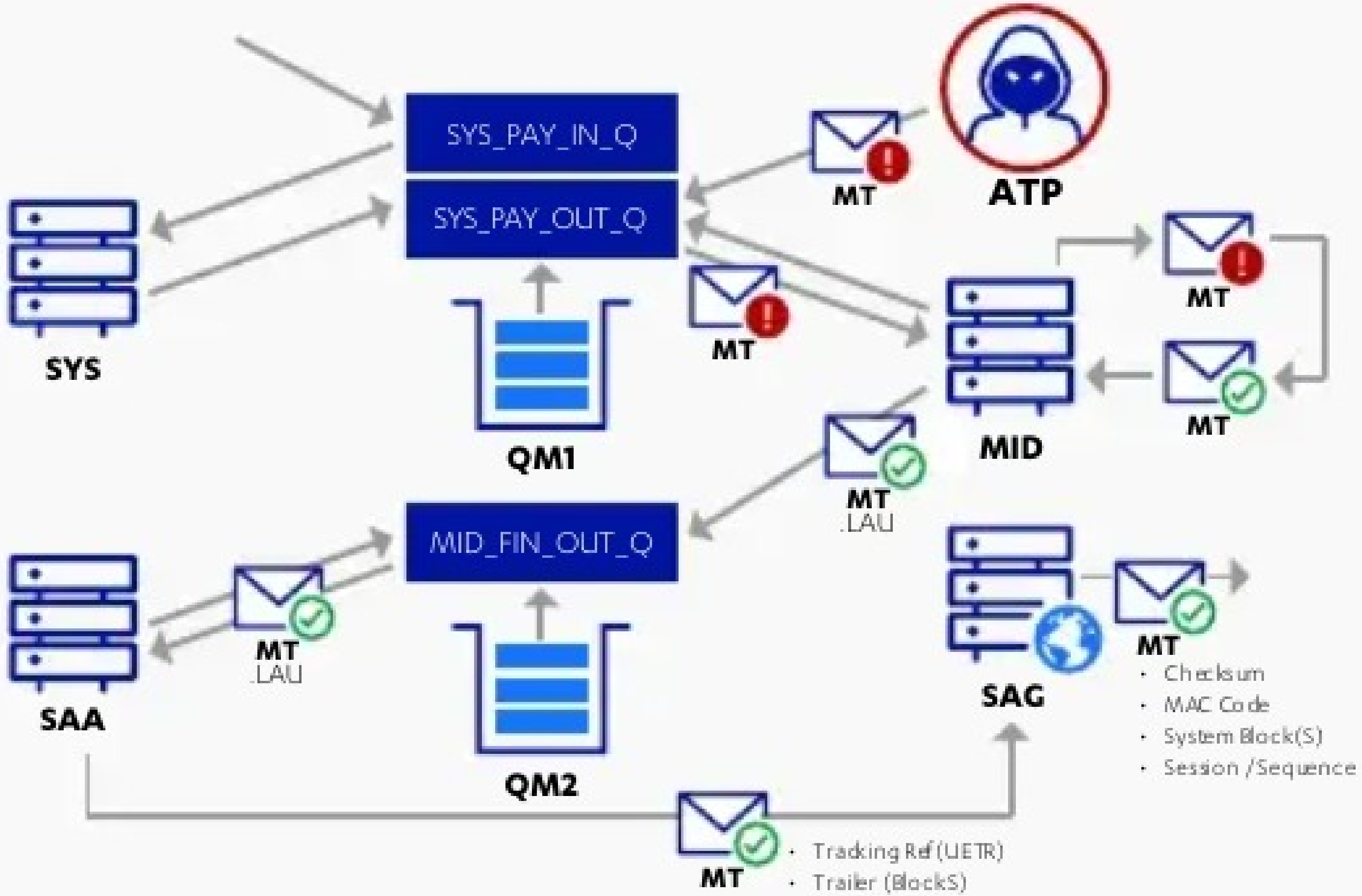


Fig. 17. Fraudulent SWIFT MT message injection

CAN YOU BE TOO SECURE FOR YOUR OWN GOOD?

In the years following CSP's initial release, F-Secure supported multiple clients with securing their SWIFT Secure Zone environments. During this time, we observed a new challenge emerging: Secure Zones were becoming highly restrictive, making it difficult to perform effective security assessments within them.

In response, we published an article describing the challenge of testing SWIFT systems in this new, highly secured world. The conclusion was that SWIFT users should be conscious that although they must restrict and harden their Secure Zone deployment, they must also implement processes and controls to allow effective security testing within those environments. This could take multiple forms, such as:

Direct access to services within the secure zone. Such direct access could be achieved through the introduction of short-lived firewall rules, allowing approved "tester" devices either restricted or unrestricted access to select systems within the Secure Zone environment, such as web application interfaces and servers for the duration of an assessment. These firewall rules could then subsequently be removed once the assessment was concluded, in order to mitigate the risk introduced during the assessment timeframe.

Access to a dedicated "testing" system connected to the Secure Zone. This could be achieved by creating a dedicated security assessment workstation with relevant tools installed that the tester accessed via Privileged Access Management (PAM) solutions. The workstation could exist within the Secure Zone or within the organization's server environment with ad-hoc whitelisted acces. As a security precaution, this system could be "powered on"—or created, if virtualized—exactly when required, mitigating any risk associated with its presence within the environment day-to-day.

Access needn't be granted to the production instance of the Secure Zone. It could instead (and ideally so) be to a pre-production instance that accurately represents the live production environment. This would facilitate the freedom to perform comprehensive testing without any risk of the assessment disrupting live traffic or exposing the production environment to unnecessary risk.

The diagram below illustrates how these high-level solutions fit into an existing SWIFT Secure Zone architecture:

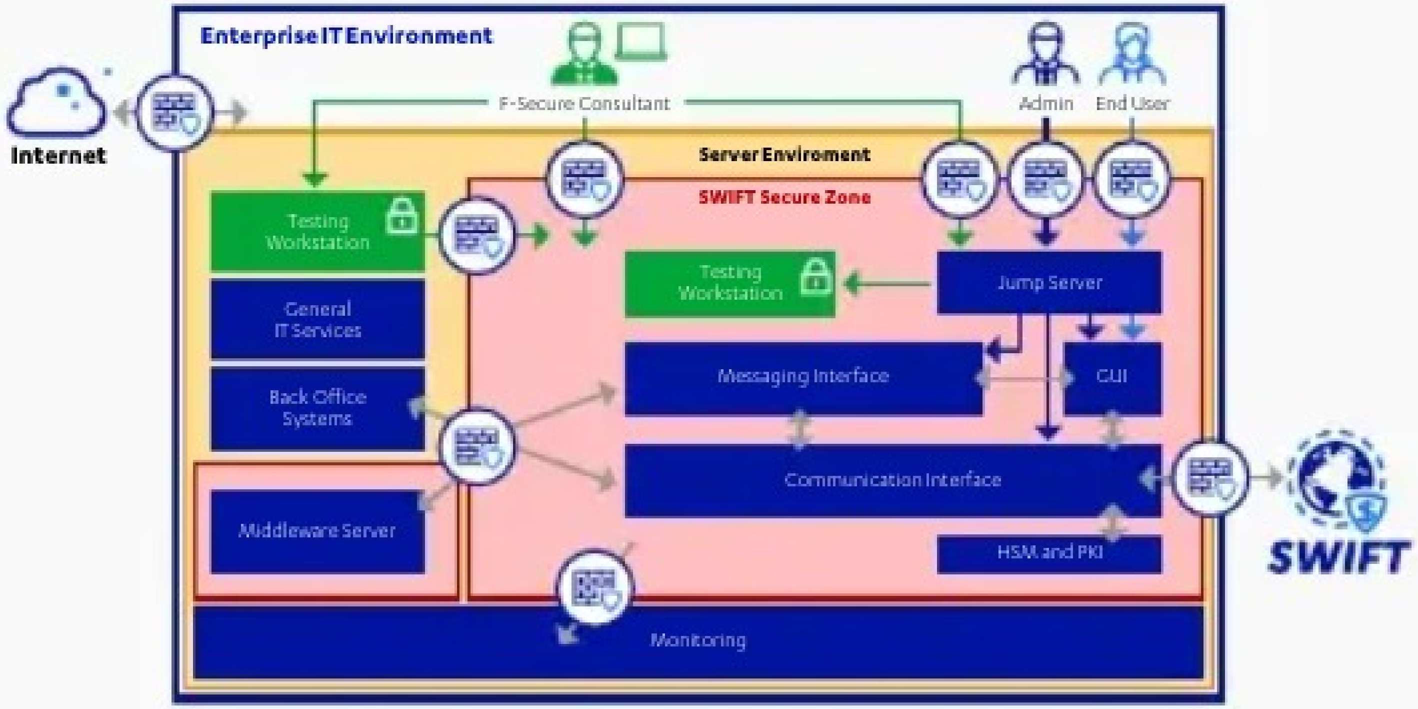


Fig. 18. SWIFT secure zone architecture

SUMMARY

F-Secure has observed multiple attacks in the wild and identified additional theoretical and practical attack paths that could be exploited by an advanced persistent threat actor to compromise an institution's local SWIFT infrastructure. A significant number of these attack paths were notably restricted following the implementation of the security controls outlined by CSP's CSCF.

SWIFT CSP and the CSCF significantly improve the security posture of an organization's local SWIFT deployment. Yet, they are often seen as a point-in-time compliance challenge focusing on only key systems, and therefore should not be relied upon alone to prevent attacks against SWIFT deployments.

Lastly, whilst creating a highly restrictive environment is effective and demonstrates a strong capability to protect critical assets, it is important to also ensure that there are systems and processes in place to review and assess these deployments.

HOW CAN YOU BETTER SECURE YOUR LOCAL SWIFT SYSTEMS?

SWIFT CSCF compliance and a complete CSP attestation will ensure and demonstrate that your local SWIFT systems are hardened and isolated within a Secure Zone. However, there will remain a number of additional systems within your wider environment that can be exploited to invoke payment instructions through SWIFT. The key aspect to recognize here is that these additional systems are not within scope of SWIFT CSP and do not reside withing the Secure Zone. As such, financial institutions must:

PREDICT

It is key that financial institutions begin by understanding and mapping out the possible attack paths that an attacker could take when attempting to compromise their enterprise network and local SWIFT infrastructure. This process begins at the SWIFT systems and works backwards towards the enterprise network perimeter in order to identify which systems communicate with the SWIFT infrastructure and the administration procedures surrounding these systems. Furthermore, all systems and applications deployed within the organization must be subject to frequent security assessments and penetration tests. A number of attempted (and successful) attacks on financial systems are never publicly reported. As such, organizations are advised to build trusted relationships with other local and international financial organizations to share information on tactics and tooling.

PREVENT

Once attack paths have been identified, an analysis of the steps an attacker would take to reach actions on objectives should be ascertained. The controls surrounding each of these steps should be assessed to confidently determine whether or not they would prevent such actions. This process should include security assessments of all controls along the path, as well as establishing an understanding of the legitimate use cases for all components.

Financial institutions should also establish a strong understanding of which systems and actions privileged users have access to, and how an attacker could subvert or abuse these privileges. If these legitimate actions are necessary and cannot be prevented, strict monitoring and detection of malicious behaviors should be implemented.

A strong focus should also be placed on establishing controls that prevent malware execution. Furthermore, these controls should be redundant in the event that one fails or is bypassed e.g.:

- **Mail gateway:** highly restrictive controls, file types limited to only those necessary, signature detection of malware, and sandbox malware detonation.
- **Endpoint devices:** anti virtus technologies should be deployed in combination with software whitelisting to prevent the execution of arbitrary binaries, scripts, and document macros.
- **Account control:** removing privileges wherever possible and adopting a "just in time/minimal effective access" approach to authentication, supported with multi-factor authentication. Privileged Access Management (PAM) platforms, can be used to centralize the management of critical system access

DETECT

Recovery from the type of cyber heists referenced through this paper is highly dependent on a timely response, facilitated by an efficient attack detection strategy. Discovering that a compromise has occurred when reading an end-of-day report is of little use; it is crucial that financial institutions implement robust logging of all key servers within the environment and maintain visibility of servers and endpoint devices through endpoint detection and response (EDR) technologies.

F-Secure further recommends that organizations adopt a threat hunting approach to detection and ensure that threat hunters are familiar with payment systems, as well as all known attacks against SWIFT systems. This should include prioritization of the endpoints (including jump hosts) that are used by privileged users, because these are the endpoints most likely to be targeted by advanced threat actors during an attack.

.

RESPOND

When prevention fails, it is these detection and response capabilities that will ultimately determine the overall financial and operational impact of an organization's local SWIFT infrastructure being compromised. Therefore, it is important that resources be given to establishing a suitable detection and response strategy surrounding your SWIFT deployment and its upstream systems. The main goal of this is to efficiently contain and recover from an attack.

Regular incident response exercises should be conducted by financial institutions to ensure that the policies and procedures in place facilitate rapid response to an incident. This should include tabletop exercises to test these procedures, as well as full incident response run-throughs based on SWIFT systems.

Attack case studies such as the heist of the Bank of Bangladesh and other major incidents should be mapped to the organization's systems. The response can then work through these to establish if an investigation could be rapidly conducted on their systems in the event of a similar attack.

CONCLUSION

In the present day cyber threat landscape, attacks on financial and SWIFT systems are still a key focus of advanced persisted threat actors. As Willie Sutton reportedly stated in 1952, when asked why he robbed banks, "that's where the money is". Huge quantities of money are for the taking. And attackers have, and continue to become, considerably more sophisticated, persistent, and resourceful.

In some of the most high-profile attacks against financial institutions, threat actors have frequently deployed bespoke malware and used advanced tactics to achieve their goal of performing fraudulent financial transactions. In response, SWIFT introduced the Customer Security Programme (CSP) and had consistently revised and updated the Customer Security Controls Framework (CSCF) to help protect its global SWIFT community from this aver adaptive threat.

However, it's important to remember SWIFT's CSP is largely a compliance challenge, which by nature becomes somewhat of a rigid, linear process. **Compliance does not ensure or imply security, as security itself is a fluid, cyclical process**, always adapting and changing. CSP might be consistely updated to adapt to this, but it will still only ever focus on the security of SWIFT infrastructure alone.

F-Secure has observed that attackers will shift their resources into targeting upstream systems and/or the users who operate with/in them. Furthermore, we've documented opportunities in the wild and researched new ways for suitably positioned attackers to successfully leverage such systems to perform a successful attack.

SWIFT's CSP recommends a number of effective hygiene measures and security controls, but the specific focus on the SWIFT payment systems will ultimately push attackers to target other parts of the organization.

As with most compromises, the root cause will frequently remain human error, whether made by administrators in a configuration file, developers in their application code, or employees being deceived into opening a malicious email attachment. For this reason, **F-Secure recommends an approach that builds on top of SWIFT CSP compliance to further strengthen the security posture of an institution as a whole**. This methodology is rooted in establishing a strong understanding of how modern threat actors target financial institutions, mapping this understanding to the organization, and selecting appropriate prevention, detection, and response measures.

A "point in time" approach to security will never succeed against an adaptive and persistent threat. The cyber threat landscape is always shifting, and so, only by turning the proposed methodology into a recurring practice can financial institutions and other organizations hope to secure themselves against future threats.

We’re global. Get in touch wherever you are.
www.f-secure.com/consulting/contact

F-Secure.

www.f-secure.com/consulting