

## CASE STUDY: WHITEBOX PENETRATION TESTING APPLIED IN ATM LOGICAL SECURITY ASSESSMENTS

Tim Eggert, Daniel Marnier  
*SRC Security Research & Consulting GmbH*

### Abstract

The move to more industry-standard and multivendor hardware architectures, operating systems and common application layer transaction protocols on Automated Teller Machines (ATMs) has made financial institutions vulnerable to similar security-related attacks as known from the regular computing environment.

Combined expertise in card payments security, ATM device driver protocol knowledge, penetration testing (certified ethical hacking) and forensic analysis allows a specific and targeted approach in assessing the vulnerability of the ATM logical security protection. Without disclosing any confidential information, the authors describe a case study where a dedicated whitebox approach was applied at the level of XFS management controls. Exploiting combined vulnerabilities and authorised by the ATM acquiring bank to inject intentionally developed program code (malware) operating on the standardised XFS interface, the authors were able to start unauthorised cash dispensing until the ATM was emptied by accessing the ATM front side as an anonymous ATM user at a self-selected date and time. The event to cash-out can be timely delayed or triggered by a predefined chipcard.

### 1. Introduction

Since many years, Automated Teller Machines (ATMs) have started to use industry-standard and multivendor hardware architectures (with USB connections for peripherals, ethernet and IP communications), operating systems and common application layer transaction protocols, introducing substantial changes in the way ATMs are deployed and installed. Increased standardisation and interoperability of ATM components reduce per unit costs, support innovation, create new service opportunities and increase flexibility and quality levels of products and services. At the same time however, financial institutions and their ATMs are becoming more vulnerable to similar security-related attacks as known from the regular computing environment. This is especially of concern in a changing context where ATMs were initially owned by financial institutions and installed in their premises, mainly on bank branch facilities, to a new reality where more and more ATMs are located off-premises and ATM acquirer networks are more often owned and controlled by independent ATM operators.

In recent years, evidence of several large-scale fraud cases on ATMs has come to the press in which criminal activities have led to penetration of ATM networks and unauthorised distribution of cash. In the best cases, such possible fraud attacks were disclosed at academic or hacker scene conferences before they could actually happen in the real world. But all cases show that attacks are becoming more and more sophisticated in terms of scale, employed technology, funding, planning and execution. ATM fraud is top of mind for all financial institutions and has made them concerned about the integrity of their ATM's software stack and their risks on financial and reputation losses.

Being well-positioned in the financial community and knowing the sensitive operations of financial institutions inside-out, SRC Security Research & Consulting provides trusted security services as an accredited evaluation lab for security assessment of payment related hardware, applications and networks. SRC is supporting financial institutions around the world in protecting from fraudulent activity and achieving compliancy with international security standards and regulations. SRC is internationally renowned and valued by financial institutions for its high-qualified expertise in assessing, designing and optimising reliable, efficient and secure payment systems that brings savings, increased security and a more professional market profile to their organisation.

The experts of SRC are highly familiar with scheme specifications of payment components and required (physical and logical) security levels. They have contributed to standardisation of ATM peripheral device drivers for accessing and manipulating the various devices of an ATM, as well have they contributed to more recent developments within XFS/IFX standardisation. Their proficiency in developing secure payment systems as well as in network and application penetration testing (certified ethical hacking) and forensic analysis surrounding

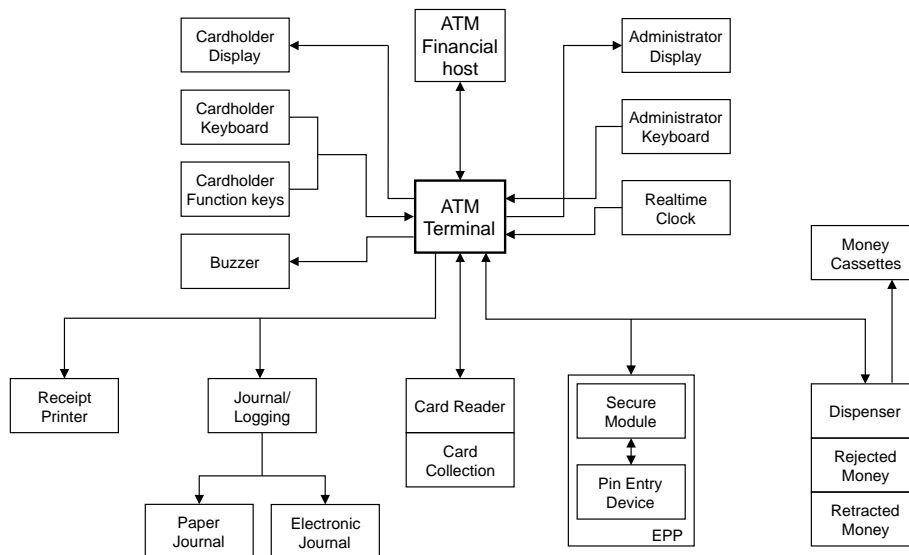
compromises of cardholder data, has allowed to develop a specific and targeted approach to vulnerability assessment of ATM and associated networks' logical security protection to known and unknown threats.

In the following the authors describe a dedicated whitebox approach applicable at the level of XFS management controls. Exploiting combined vulnerabilities and the injection of intentionally developed program code (malware) operating on the standardised XFS interface, the authors verified in a case study that they were able to start unauthorised cash dispensing until the ATM was emptied by accessing the ATM front side as an anonymous ATM user at a self-selected date and time. The event to cash-out can be timely delayed or triggered by a predefined chipcard.

Mitigation of risks presented in this document aim to reduce such possible fraud attacks.

## 2. General ATM functional overview and scope of assessment

The following picture shows a functional overview of the physical peripheral parts of an ATM:



The installed software includes the (hardened!) operating system, the ATM application (end-user application as well as the administration applications, for example, application software downloading, operating system updates, monitoring of the ATM, etc), the peripheral device drivers and the firmware (hardware intimate code). Usually, mechanisms are in place to detect alteration of the software. As all of the peripherals are under control of the ATM, they are all likely candidates for attacks. Vulnerability assessment and penetration testing are focused on the possibilities of manipulating the (arrow-lined) interfaces, with the aim to determine if unauthorised access or other malicious activity could lead to unauthorised distribution of cash. Typical penetration testing includes network and application layer testing as well as controls and processes around the networks and applications.

The CEN XFS Standard<sup>i</sup> provides a common API for accessing and manipulating the various peripheral devices of an ATM. Access to XFS commands should be restricted through the access control mechanisms of the ATM operating system. As different interpretations of the XFS standard exist, typically, a middleware layer is used to even out the differences between various platforms. This middleware layer is usually the result of a proprietary development by the ATM manufacturer, often requested by ATM operators to have specific functionality included for their specific purpose and therefore more vulnerable to attacks exploiting weaknesses in this management layer.

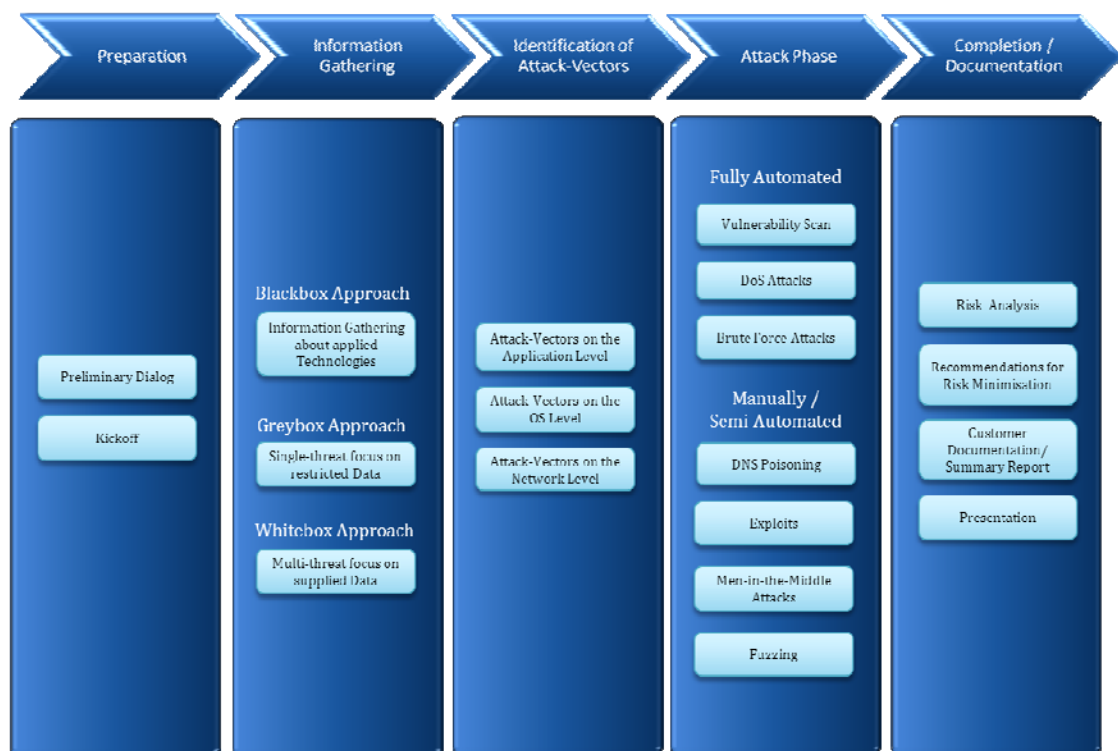
<sup>i</sup> ref.: <http://www.cen.eu/cen/Sectors/Sectors/ISSS/Activity/Pages/WSXFS.aspx>

### 3. General SRC Approach to Penetration Testing<sup>ii</sup>

SRC takes the following documents as a baseline in performing penetration testing:

- *Durchführungskonzept für Penetrationstests*, BSI (Bundesamt für Sicherheit in der Informationstechnik, German Federal Office for Information Security)
- *Ethical Hacking and Countermeasures Guidelines*, European Union
- Several Industry Best Practices

These documents have been the foundation for an SRC developed Penetration Testing Approach in which the following phases and approaches are defined:



Referring to the level of knowledge provided on systems, architectures, networks, applications, operating systems, procedures, access mechanisms, etc., initial analysis and actual testing can be performed on the basis of three different approaches mentioned in the picture above:

- a *Blackbox* approach, accurately simulating an actual “outsider” hacker with no or limited prior knowledge on any of the above items. Publicly available information (via for instance queries on search engines, mailing lists, newsgroups, public databases and other internet sources) is collected to find out about known malware and exploits. Also, passive and active assessment of environmental information through e.g. packet-sniffing, portscans, OS and application fingerprinting, service identification, desktop firewall evasion, IPS/IDS detection and network mapping is used to find out about potentially existent vulnerabilities.
- a *Greybox* approach, with a narrowed down scope and with a dedicated focus on a single threat scenario with a single entry point at certain suspected parts of the ATM. This approach can be useful in case the ATM operator wants to receive more detailed investigations on a specific (e.g. known) “insider” attack, without already disclosing all available (and mostly confidential) information. Reverse engineering (if not restricted by local laws<sup>iii</sup>) as a method to understand the application logic can be part of this.

<sup>ii</sup> Penetration testing needs to be performed in accordance with critical company processes including change control, business continuity, disaster recovery, operational safety and further security policies

<sup>iii</sup> Some countries have legal restrictions for performing software reverse engineering activities in the country where the analysis is performed

- a *Whitebox* approach, accurately simulating an actual “insider” hacker attacking with inside knowledge up to the level of information that bank branch staff, IT staff or ATM manufacturer technical staff has. A *Whitebox* approach assumes knowledge of e.g. manufacturer product documentation, interface descriptions and protocol information, applications, programming languages, software development life cycle, privilege management, logging mechanisms, information on security and coding policies (e.g. hardening guidelines) and patch management, manufacturer test- and debugging tools, source codes, access to manufacturer technical staff, etc. In addition and if not fully and securely encrypted, a core ATM storage device can be investigated via forensic analysis (on a virtualised and secure forensic copy of the original data) to mimic the situation of an attacker possessing an ATM device after theft with the aim to assess which confidential or critical data (e.g. card data, customer data, login credentials, etc) is (left) available.

The increasing amount of information provided in the above three approaches is directly proportional to the probability of success in attacking the ATM. All approaches aim to gain detailed knowledge of transport and application layer protocols on both internal and external interfaces and in case a vulnerability is detected and only after explicit agreement of the ATM operator or acquiring bank customer, specific program code (malware) can intentionally be developed that is used to prove the vulnerability concept.

In the attack phase all identified attack vectors are applied. Before starting the activities and attempting any exploits, explicit agreement with the customer will be required on any limitations in allowable methods and techniques. SRC distinguishes between fully automated and semi-automated or manual attacking.

A fully automated attack is performed on the basis of tools configured with acquired data and identified attack vectors, without direct involvement of the evaluator(s). For this purpose tools like Nessus, AppScan, Metasploit, Burp Suite, WebScarab and other, more specialised tools are used.

A semi-automated or manual attack requires a high direct involvement of the evaluator(s) and aims to simulate an advanced attacker. Attacks will be carried out on the integrity and availability (Denial of Service) of the systems to test their performance and resistance. Attacking takes place on the basis of custom-designed penetration tests and a procedure with scope and timing that has been revised and agreed upon with the customer and communicated to affected parts of the organisation.

Exemplarily, the following methods can be used in an attack:

- The attacker will try to build, change and/or delete data in/from databases or transmit queries to databases that return more data than the system would do when working correctly.
- Session-hijacking attacks where the attacker will try to take-over user-sessions and see, change, falsify or delete any user-data.
- SSL man-in-the-middle attacks: If an attacker successfully carries out an SSL man-in-the-middle attack, he can sniff data into the encoded tunnel, see, change and try to falsify it, in case such decoded login data and sensitive information are transmitted within the tunnel.
- Test on replay possibilities where it is tested to see if the data from one client can be recallable through another client. If procurable, an attacker can see or change the data from different clients.
- Information disclosure: provocation of error messages to gain version details from services. This provides sometimes useful and detailed information about the server services and their configuration, which at the same time gives information about security vulnerabilities, consequently allowing targeted attacks.
- Tests of buffer or heap overflow (POST) attacks whereby a DoS can be caused by shutting down or freezing the server service. If applicable, it will be possible to open a so called “root-shell” and gain complete administrative access to the system, add additional users or compromise the system in another way (e.g. by installing keyloggers etc.).
- Spoofing and fuzzing: typically applied in a *Blackbox* approach to find vulnerabilities by manipulating protocols, files, etc.
- Apply brute force methods or otherwise exploit weak encryption
- Based on previous research or acquired documentation, the attacker will determine and use vulnerabilities through analysis and investigation of interfaces and protocols.

#### 4. Case Study: a real world Whitebox Approach in ATM Vulnerability Assessment

Within a case study SRC experts assessed the vulnerability of ATM installations in a *Whitebox* approach with the specific objective of trying to produce a cash-out of the ATM without being detected by the ATM implemented security services.

The analyzed ATM base was (among others) armed with the following defenses:

- Maximum security settings with a hardened operating system (i.e. all unnecessary services were shut down, all unnecessary ports were closed, etc.)
- All recent security patches and service packs were applied
- Stringent firewall settings, locking all electronic entry points by monitoring, analysing and authenticating any external source attempting to connect, blocking anything the software does not recognise and sending alert messages upon detecting unauthorised activity

The following methodology was applied (obeying the baseline documents for penetration testing as explained before):

- Research on the ATM configuration and the CEN XFS SDK
- Network identification of the ATM and identification of possible network based attacks
- Forensic analysis on a virtualised and secure forensic copy of a core ATM storage device
- Identification of the hardware drivers controlling the Cash Dispenser
- Investigate possible reverse engineering of the hardware drivers
- Identification of dependencies within the software layers
- Identification of function calls accessing these drivers (and their appropriate order) within the software layers
- Identification of memory protection mechanisms
- Based on detected vulnerabilities, create a Proof of Concept program code to trigger the cash-out

Research was for a large part focused on the CEN XFS standard and more specific on the SDK provided by CEN for accessing XFS. Network identification and forensic analysis together revealed around 11 vulnerabilities of which 3 vulnerabilities proved crucial for the next phase, emerging as successful attack vectors. Network traffic capture and analysis revealed a medium risk vulnerability about the remote desktop protocol in use. Without disclosing the precise nature of this vulnerability in this document, this vulnerability was used later on to bypass the maintenance mode restrictions. This made it possible to download a malware into the ATM without being noticed by the system. A second vulnerability exposed a C# remoting service that could help to remotely deploy malware.

Based on the CEN SDK a cash-out Proof of Concept program was developed. Necessary adoptions in the beginning included the move from SDK 4.0 (Visual Studio 2010) to SDK 3.5 (Visual Studio 2008) as well as decision to write the PoC in the programming language C instead of C++ or C#, which tended to have memory allocation issues. Much time during development had to be spent on setting up the correct data structures for the XFS commands as well as setting up the XFS communication. The developed PoC was completely evading the overall ATM application and communicating directly to the XFS Manager running on the ATM. Since the XFS used no authorisation at this point, the cash-out command was accepted and resulted in a successful cash delivery. It proved not necessary to reverse engineer any vendor specific drivers. Furthermore, investigations of attack paths after discovery of a maintenance access vulnerability during PoC development have shown multiple possibilities to access the file system without any knowledge of login credentials, once physical access to the ATM's internals is available. Such access would allow installation of unauthorised program code (malware) by e.g. maintenance engineers.

The PoC was afterwards modified to allow residing in the background during normal operations of the ATM and automatic startup at insertion of a specific chipcard with dedicated ATR (Answer To Reset) in order to trigger cash dispensing. Due to the XFS standard being implemented on various manufacturer's ATMs, such a malware could easily be adapted to other ATM vendors as target platform.

## 5. Mitigation of risks

To achieve an in-depth protection, various aspects of the whole ATM system have to be taken into account. A rough outline of the possible attack paths for physical and network access will help to understand the necessary mitigations.

In order to harden the ATM against attacks, generic system hardening should be applied in the first place. This would mean e.g. applying security patches for the operating system in time and allowing only a minimum of needed services to listen on any outbound network interfaces.

In order to avoid unauthorised cash-out via XFS, the execution of 3<sup>rd</sup> party code should be prevented. Blocking of external storage devices via USB (to prevent introducing malware) is normally difficult due to maintenance tasks and business needs. Since all attempts to secure the USB ports can be circumvented by the disconnection of a legitimate device and the replacement with an USB storage, it would be recommended to remotely monitor all USB activities (especially device disconnects) and administrative tasks. This should also include any actions involving other external storage solutions like CD or Firewire (IEEE 1394) devices. This monitoring cannot prevent the introduction of custom malware, but can help to detect unauthorized modifications.

Furthermore, it should be evaluated, to what extent all XFS transactions can be logged. In addition, a recheck in the maintenance access disclosed that the amount of banknotes (counted by the maintenance application) had NOT changed. It is therefore recommended to monitor the amount of banknotes on the lowest application or hardware level possible.

As a final recommendation, best business practice currently is to use the central systems' network administration and security policies for authentication and authorisation of users, assigning and/or enforcing security policies and installing or updating software. This helps to reduce the available functions to a minimal set of programs needed for daily use.

## 6. More information

SRC  
Security Research & Consulting GmbH

Graurheindorfer Straße 149a  
D-53117 Bonn – Germany

Telephone: +49(0)228/2806-100  
Telefax: +49(0)228/2806-199

Email: [info@src-gmbh.de](mailto:info@src-gmbh.de)  
Internet: [www.src-gmbh.de](http://www.src-gmbh.de)

