POSITIVE TECHNOLOGIES

ATM LOGIC ATTACKS: SCENARIOS

2018

Contents

Introduction

In January 2018, the U.S. Secret Service, as well as major ATM vendors Diebold Nixdorf and NCR, issued urgent warnings about the threat of attacks on ATMs. These warnings were notable because of the nature of the threat: criminals were said to be planning to plant malware on ATMs or connect special devices to control cash dispensing.

A few months earlier, in October 2017, a series of such attacks had occurred in Mexico. The attackers specially prepared a malware-laden hard drive in advance and switched it with the ATM's original hard drive. To restore the connection with the cash dispenser in the ATM, the attackers emulated physical authentication, which is needed to confirm that authorized access to the ATM's internal safe has been obtained. With the help of a medical endoscope, the attackers succeeded in manipulating the dispenser sensors. According to NCR reports, Black Box attacks were also recorded during this same period. Instead of switching out the ATM hard drive, the attackers connected a special device (a "Black Box") to send commands to the cash dispenser, from which cash was then collected by the attackers. In January 2018, these attacks spread to the United States.

What these incidents had in common was that, instead of trying to physically pry cash out of ATMs, the attackers emptied their targets with the help of malware or special hacking devices. Such logic attacks require greater technical skill and preparation, but reward criminals with a quiet method of theft that brings a lower risk of being caught.

While logic attacks are a newcomer to the United States, they have long plagued the rest of the world. The first reports of in-the-wild ATM malware came in 2009, with the discovery of Skimer, a Trojan able to steal funds and bank card data. Ever since, logic attacks have become increasingly popular among cybercriminals. This trend is underscored by the European Association for Secure Transactions (EAST) report on 2017 ATM attacks. Compared to 2016, the number of logic attacks in Europe tripled in 2017, with total damages of €1.52 million.

Skimer, used in the very first attacks, is still under active development today. Other malware families—including GreenDispenser, Alice, Ripper, Radpin, and Ploutus—have appeared as well. All these are available on darkweb forums. With prices starting at $1,500, such malware is relatively expensive. But the potential profits are enormous. Attackers can recoup their initial costs with even one successful theft. Meanwhile, malware developers are adapting their "products" to an ever-growing variety of ATM models. CutletMaker malware, first spotted in 2017, was sold openly together with detailed instructions for a price of $5,000.

The most important thing about ATM malware is not its inner workings, but the installation method. The first step for protecting banks and their clients is to identify potential infection vectors and vulnerable components. In this report, we will share the results of ATM security analysis performed by our company in 2017–2018, discuss different types of possible logic attacks identified during such work, and provide recommendations for securing ATMs.

Client snapshot

Our sample consists of the 26 ATMs for which we performed maximally complete security analysis during the time period in question. These ATMs were manufactured by NCR, Diebold Nixdorf, and GRGBanking. Each ATM had a unique configuration. The range of attacks possible against any single model varied widely depending on the type of connection to the processing center, software installed, security measures in place, and other factors. The following table provides an overview of characteristics for these ATMs.

Table 1. Configurations of tested ATMs

ATM model	OS version	Application Control	Protection against Black Box attacks	Processing center connection type
GRGBanking H68NL	Windows 10	KXSecurity	Authentication and encryption of data between OS and dispenser	Direct connection
NCR Personas 6676	Windows XP	McAfee Solidcore	NCR USB Encryption Level 3	Direct connection or hardware VPN client
NCR SelfServ 5877 (Configuration 1)	Windows XP	N/A	N/A	Hardware VPN client
NCR SelfServ 5877 (Configuration 2)	Windows XP	McAfee Solidcore	NCR USB Encryption Level 3	Direct connection or hardware VPN client
NCR SelfServ 6622 (Configuration 1)	Windows 7	McAfee Solidcore	NCR USB Encryption Level 3	Direct connection
NCR SelfServ 6622 (Configuration 2)	Windows 7	McAfee Solidcore	NCR USB Encryption Level 3	Software VPN client
NCR SelfServ 6622 (Configuration 3)	Windows 7	Windows AppLocker	NCR USB Encryption Level 3	Software VPN client
NCR SelfServ 6622 (Configuration 4)	Windows 7	Windows AppLocker	NCR USB Encryption Level 3	Software VPN client
NCR SelfServ 6622 (Configuration 5)	Windows 7	McAfee Solidcore	NCR USB Encryption Level 3	Direct connection
NCR SelfServ 6622 (Configuration 6)	Windows 7	McAfee Solidcore	NCR USB Encryption Level 3	Software VPN client
NCR SelfServ 6622 (Configuration 7)	Windows XP	McAfee Solidcore	NCR USB Encryption Level 3	Software VPN client
NCR SelfServ 6622 (Configuration 8)	Windows XP	McAfee Solidcore	NCR USB Encryption Level 3	Direct connection or hardware VPN client
NCR SelfServ 6632	Windows XP	McAfee Solidcore	NCR USB Encryption Level 3	Hardware VPN client
NCR SelfServ 6683	Windows 7	McAfee Solidcore	NCR USB Encryption Level 3	Direct connection or hardware VPN client
NCR SelfServ 6822	Windows XP	GMV Checker ATM Security	NCR USB Encryption Level 3	Hardware VPN client
WN 2000	Windows XP	M3.Defender	N/A	Direct connection or hardware VPN client
WN 2000XE (Configuration 1)	Windows XP	M3.Defender	N/A	Direct connection or hardware VPN client
WN 2000XE (Configuration 2)	Windows XP	M3.Defender	Cerber Lock	Direct connection or hardware VPN client
WN 2000XE (Configuration 3)	Windows XP	SafenSoft	N/A	Direct connection or hardware VPN client
WN 2000XE (Configuration 4)	Windows XP	SafenSoft	Wincor USB Encryption	Direct connection or hardware VPN client
WN 2100XE (Configuration 1)	Windows XP	N/A	Wincor USB Encryption	Direct connection or hardware VPN client
WN 2100XE (Configuration 2)	Windows XP	SafenSoft	Cerber Lock	Direct connection or hardware VPN client
WN 2100XE (Configuration 3)	Windows XP	Symantec PC/E Terminal Security	N/A	Direct connection or hardware VPN client
WN C4040 (Configuration 1)	Windows 7	M3.Defender	Wincor USB Encryption	Direct connection or hardware VPN client
WN C4040 (Configuration 2)	Windows 10	Kaspersky KESS	Wincor USB Encryption	Hardware VPN client
WN C4040 (Configuration 3)	Windows 10	SafenSoft TP Secure	Wincor USB Encryption	Hardware VPN client

How an ATM works

Before looking at attack scenarios, let's first get a better idea of what an ATM is and which ATM components might be of interest to attackers.

An ATM consists of two main parts: cabinet and safe. The cabinet (main body) contains the ATM computer, which is connected to all the other devices: network equipment, card reader, keyboard (PIN pad), and cash dispenser (the dispenser itself is in the safe, but the connector is not). The cabinet is practically unprotected, with only a plastic door secured by a trivial lock. What's more, manufacturers usually use the same lock for all ATMs of the same series. Keys for these locks can be purchased easily online, although attackers can also pick them or drill through the flimsy plastic. The safe is more robust, being made of steel and concrete, and contains only the cash dispenser and cash acceptance module.

The computer usually runs on Windows, in a special embedded version designed specifically for ATM use. Only administrators should have access to Windows; other users should not have such access. This is why user-facing applications run in kiosk mode. These applications provide all necessary functionality to the user: this is the interface that we see during normal ATM use.

To do its job, the application must communicate with ATM peripherals: get card information from the card reader, obtain user input from the keyboard, and send commands to the cash dispenser. This communication takes place using XFS (extensions for financial services), a standard for simplifying and centralizing equipment control. With XFS, a hardware manager makes an API available to all Windows applications and forwards requests to devices. Commands to each XFS-connected device are sent via the corresponding service provider (device driver). The hardware manager translates API functions to SPI functions and forwards the result to the service providers. Each ATM vendor implements XFS in their own way.

An ATM never decides to dispense cash all by itself. When processing a transaction, it contacts the bank's processing center. This connection is either wired or wireless (for example, via a mobile data network). It is important to secure the connection against data interception. In most cases, software or hardware VPN clients perform this task.

Data exchange with the processing center most often occurs via the NDC or DDC protocols, although banks sometimes use their own methods. Besides the processing center, the ATM is also connected to the bank's internal network (for remote administration) and software update server.

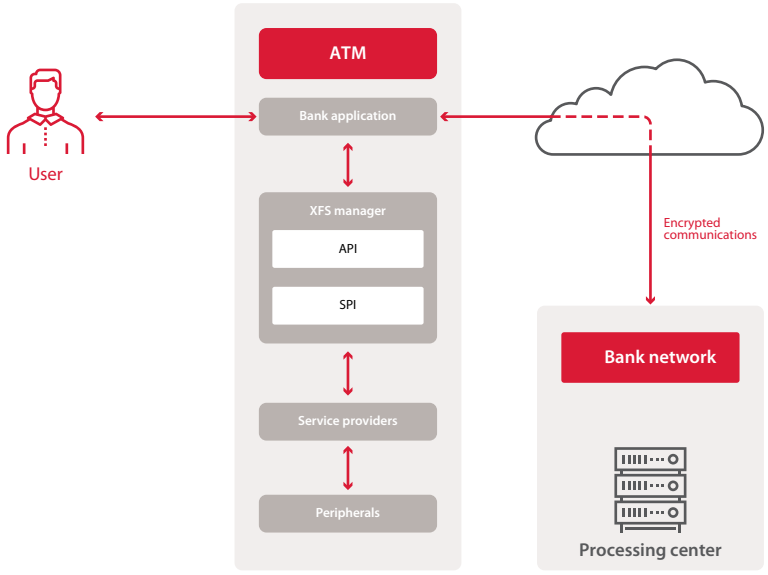
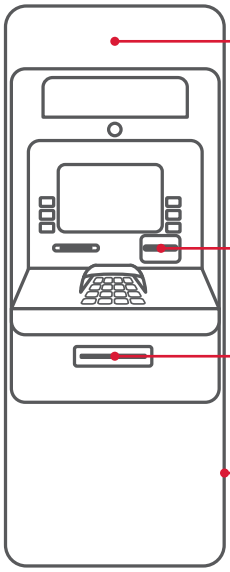


Figure 1. Interactions between ATM components

For criminals, the interesting parts of an ATM include the computer, network equipment, and main peripherals (card reader and cash dispenser). An attack on these components could enable intercepting card data, interfering with transaction processing by the processing center, or telling the dispenser to issue cash. For such attacks, the criminal requires physical access to the cabinet of the ATM or a connection to the network on which the ATM is located.



	Device	Possible attacks
	ATM computer	Connection of USB devices Connection to hard drive Boot mode modification
	Card reader decodes data from bank cards	Card data interception
	Cash dispenser issues banknotes from the safe	Black Box
	Network equipment connects the ATM to the processing center and remote administration servers	Interception of network traffic Spoofing of processing center Attack on available network services Attack on network devices Connection from the bank's internal network Spoofing or hacking of software update server

Figure 2. Possible attacks on ATM devices

Vulnerability types

ATM vulnerabilities encountered in security analysis fall into four categories:

- Insufficient network security
- Insufficient peripheral security
- Improper configuration of systems or devices
- Vulnerabilities or improper configuration of Application Control

In the case of insufficient network security, a criminal with access to the ATM network can target available network services, intercept and spoof traffic, and attack network equipment. Criminals can also spoof responses from the processing center or obtain control of the ATM. Tested ATMs frequently featured poor firewall protection and insufficient protection for data transmitted between the ATM and processing center.

In many cases, the cause of insufficient peripheral security is lack of authentication between peripherals and the ATM OS. As a result, a criminal able to infect the ATM with malware can access these devices or directly connect their own equipment to the dispenser or card reader. The criminal can then steal cash or intercept card data.

Improper configuration refers to gaps in protection that a criminal can abuse if able to obtain access to the cabinet of the ATM: lack of hard drive encryption, authentication errors, poor protection against exiting kiosk mode, and the ability to connect arbitrary devices.

The fourth category consists of Application Control vulnerabilities. Such solutions are intended to prevent execution of unwanted code on the ATM, but in practice often fail to live up to their billing. Vulnerabilities may lurk in Application Control code or result from improper configuration.

In the following section, we will detail the vulnerabilities discovered by our experts and related potential attack scenarios successfully demonstrated during testing.

Attack scenarios

We have divided attack scenarios into two categories, based on their objective: obtaining money from the ATM safe or copying information from clients' bank cards.

Cash robbery

Network attacks

Vulnerable to attack	Access needed	Time needed
85% of ATMs	Access to ATM network	**15** minutes

For network-level attacks, the main requirement is access to the network to which the ATM is connected. If the attacker is an employee of the bank or Internet provider, this access can be obtained remotely. Otherwise, an attacker needs to be physically present to open the ATM, unplug the Ethernet cable, and connect a malicious device to the modem (or replace the modem with such a device). Then it is possible to connect to the device and attack available network services, or attempt man-in-the-middle attacks. Sometimes the modem is located outside of the ATM cabinet, so an attacker would not even have to open up the ATM in order to perform modifications.

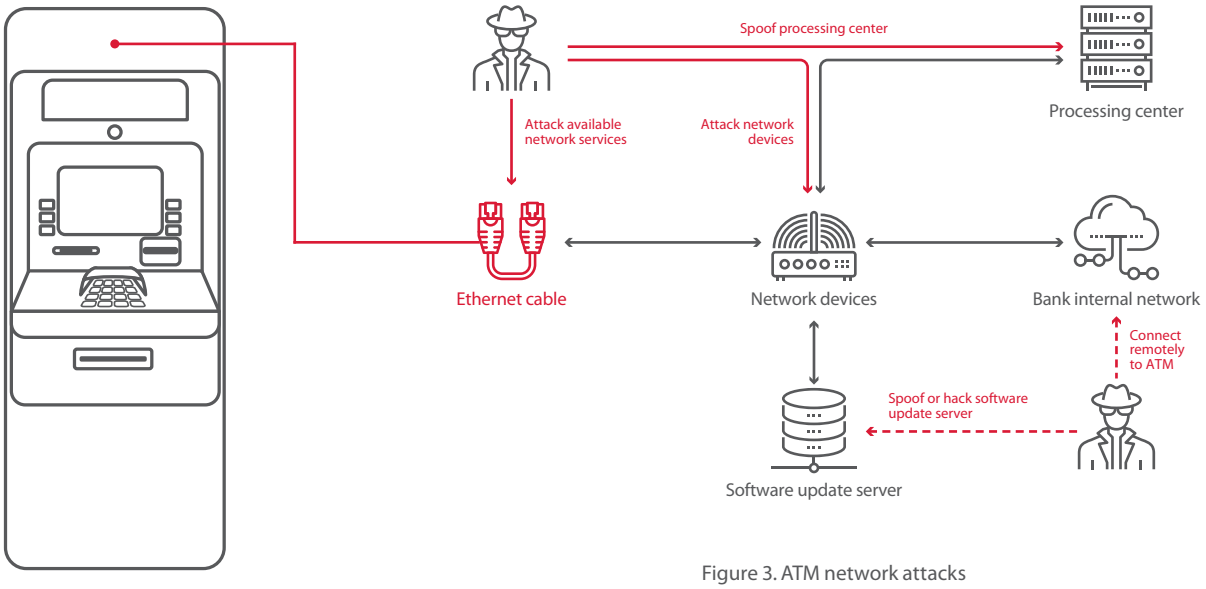


Figure 3. ATM network attacks

Here we will not delve into attacks that involve hacking bank IT infrastructure. Suffice it to say that an attacker able to penetrate a bank's internal network will also obtain access to ATM management and the ability to place malware on ATMs. This was the modus operandi of the Cobalt group, for example. In early 2018, we reported our findings on the security of bank information systems: our experts were able to obtain unauthorized access to ATM management at 25 percent of tested banks.

Spoofing of processing center

At risk:
27% of tested ATMs

If data between the ATM and processing center is not secured, an attacker can manipulate the transaction confirmation process. A processing center emulator approves any request received from the ATM and, in its response, sends a command to dispense cash. The emulator is connected via Ethernet cable to the ATM cabinet or replaces network equipment.

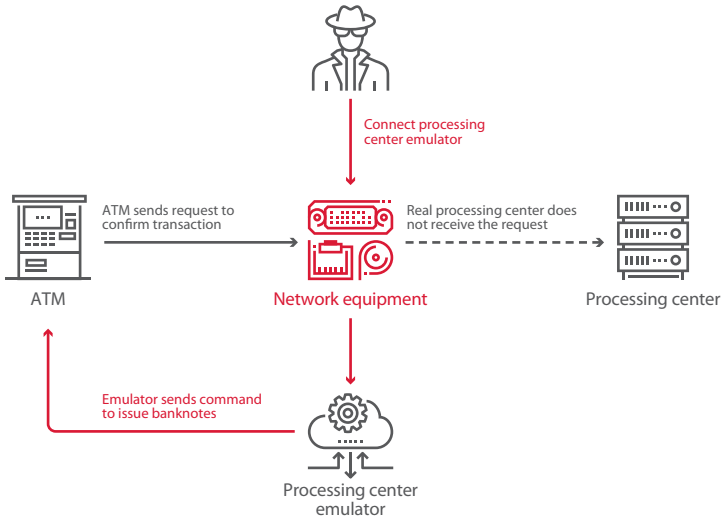


Figure 4. Spoofing the processing center

Spoofing the processing center is possible when three conditions are met simultaneously:

- **Data between the ATM and processing center is not specially encrypted.** Since the NDC and DDC protocols do not natively employ data encryption, an attacker can intercept and modify information.
- **VPN protection is poorly implemented.** On the ATMs we tested, both software and hardware VPN solutions could be disabled. If the VPN client is placed outside of the ATM, or if the attacker can access the ATM cabinet, the attacker can install their own equipment between the ATM and VPN hardware.
- **Message Authentication Codes are not used in transaction requests and responses**, which enables altering traffic without being detected.

During testing, experts identified another attack scenario in which responses from the processing center could be faked. ARP Spoofing is a man-in-the-middle attack in which changes are made to an ARP table by sending fake ARP Response messages. As a result, traffic is redirected via the attacker's equipment. If traffic is not encrypted, the attacker can alter the contents of a response, such as by increasing the number of banknotes to dispense.

```
Scanning for merged targets (2 hosts)...
* |==================================================>| 100.00 %
2 hosts added to the hosts list...
ARP poisoning victims:
 GROUP 1 : 192.168.13.1 E0:CB:4E:48:E6:B1
 GROUP 2 : 192.168.13.105 00:1A:D4:21:45:4A
```

Figure 5. Demonstration of ARP Poisoning attack

Figure 6. Spoofing of response from processing center (command to dispense one banknote)

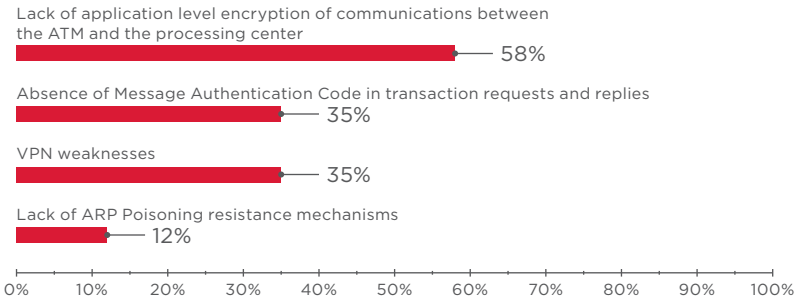


Figure 7. Incidence of vulnerabilities (percentage of tested ATMs affected)

At risk:
58% of tested ATMs

Exploitation of vulnerabilities in available network services

An attacker can exploit vulnerabilities in available network services, including remote control services, and thereby execute arbitrary commands. Consequences include disabling security mechanisms and controlling output of banknotes from the dispenser.

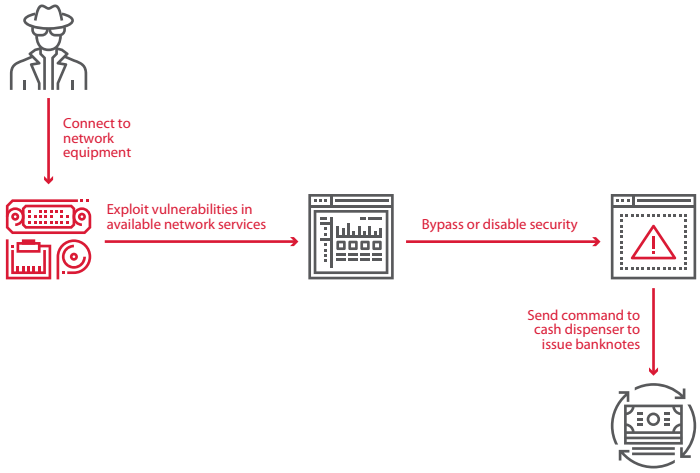


Figure 8. Exploiting vulnerabilities in available network services

Vulnerabilities needed for this attack vector are caused by poor firewall protection, use of vulnerable or out-of-date software versions (for example, vulnerabilities CVE-2017-8464 and CVE-2018-1038 enable remotely running arbitrary code and subsequently escalating privileges), and improper configuration of security tools (application whitelists tend to be excessively generous, as detailed later in this report).

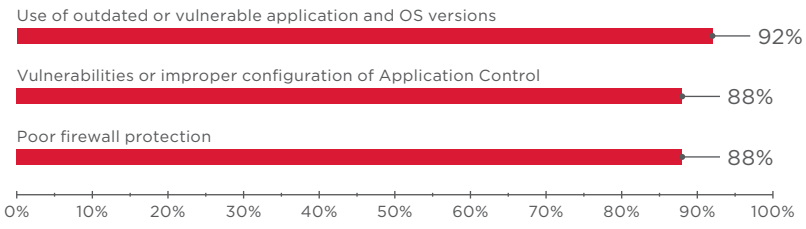


Figure 9. Incidence of vulnerabilities (percentage of tested ATMs affected)

At risk:
23% of tested ATMs

Attacks on network devices

Another way to obtain network access is to target the network devices connected to the ATM. Attackers can take control of equipment and then start targeting other ATMs on the same network, and even the bank IT infrastructure.

Here is an example encountered by our experts during one project. They were analyzing the firmware of a GSM modem used to create a mobile data network. The network handles traffic with the processing center, video materials, event notifications, and remote access to the ATMs. Hosts on the network can communicate with each other using a special protocol. This protocol supports special messages such for getting information about a host, reading configuration files, and running OS commands.

Message traffic is encrypted using a session key, which is generated based on the host key. The host key, in turn, is encrypted based on yet another key stored in the modem firmware. An attacker with physical access to the modem can read the firmware with the help of special hardware and software. During testing, the experts extracted the key from the firmware and connected to the network.

The configuration files of hosts on the network contained the addresses of servers on the internal bank network. These servers were accessible from the mobile data network being tested and supported the protocol messages already mentioned, such as for running OS commands. So by obtaining the key from the modem firmware, an attacker could take control of the bank's internal infrastructure. The testers were able to advance the attack to obtain access to payment gateways, databases, and video servers.

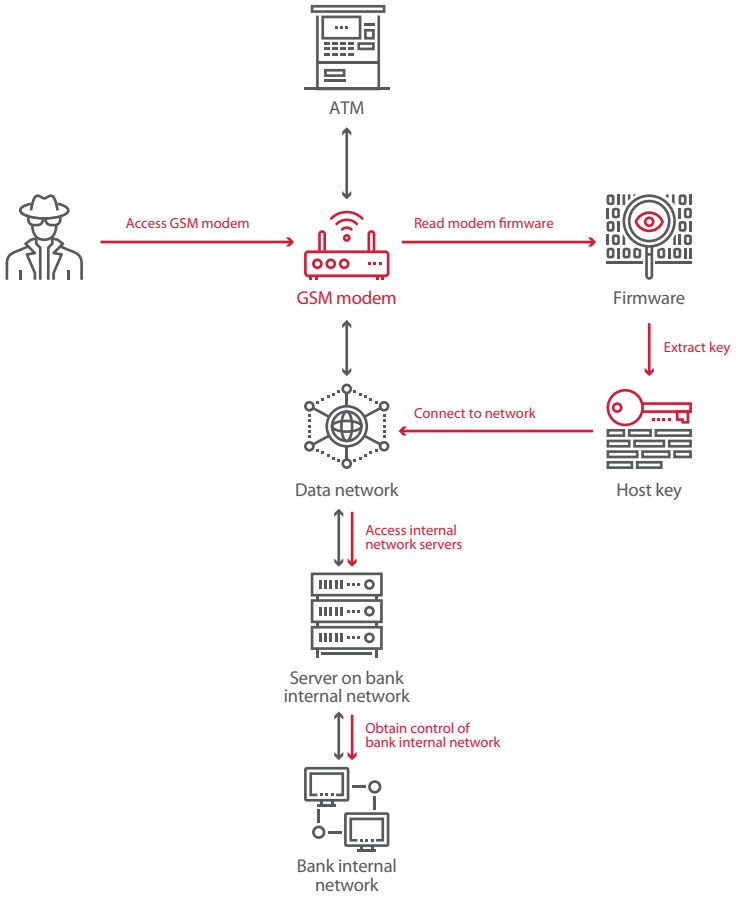


Figure 10. GSM modem attack scenario

In another project, the potential attack vector involved the fact that after installation of a GSM modem, network interfaces for remote administration remained open and default credentials were used.

The experts connected the GSM modem to their own (fake) base station. They then discovered two open network interfaces: Telnet and the web administration interface. The comically insecure combination root:root was set on the device, which enabled quickly obtaining access with maximum privileges via Telnet. Weak credentials were bruteforced for the web interface as well. An attacker could use this to direct network traffic to a malicious device, intercept requests, and spoof responses from the processing center.

Recommendations

1. Place network equipment inside the ATM.
2. Use a software or hardware VPN client located inside the ATM.
3. Use strong encryption for data between the ATM and processing center.
4. Include a Message Authentication Code in all transaction requests and responses.
5. Secure or disable unused link-layer and network protocols.
6. Configure the firewall to allow remote access only to services required for ATM operation. Close all network interfaces to which access is not needed. Remote access should be allowed only from whitelisted administrator addresses.
7. Enforce a strong password policy for remote control access.
8. Regularly install operating system and application updates.
9. Log and monitor security events.

Black Box

Vulnerable to attack	Access needed	Time needed
69% of ATMs	Physical access to ATM cabinet	**10** minutes

As mentioned already, the cash dispenser is located within the safe, which is physically well protected. But the connection of the cash dispenser to the ATM computer is located outside of the safe, and therefore easy to access. In some cases, criminals have drilled holes in the front panel of an ATM in order to access the dispenser cable. With such access, criminals can then directly connect the dispenser to their own device, which is programmed to send cash dispensing commands. This device is most often a simple single-board computer (such as Raspberry Pi) running modified versions of ATM diagnostic utilities. Diagnostic utilities usually run checks to verify that access is legitimate, but attackers know how to disable these checks and any other security mechanisms. These techniques are combined in what are known as Black Box attacks.

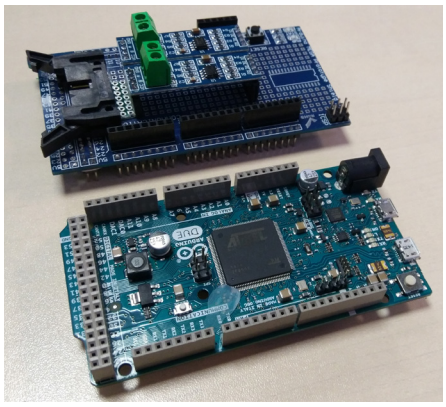


Figure 11. Black Box components

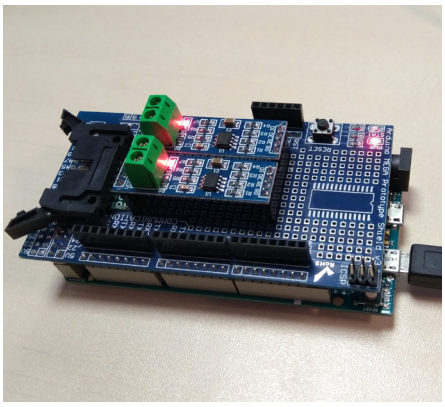


Figure 12. A Black Box

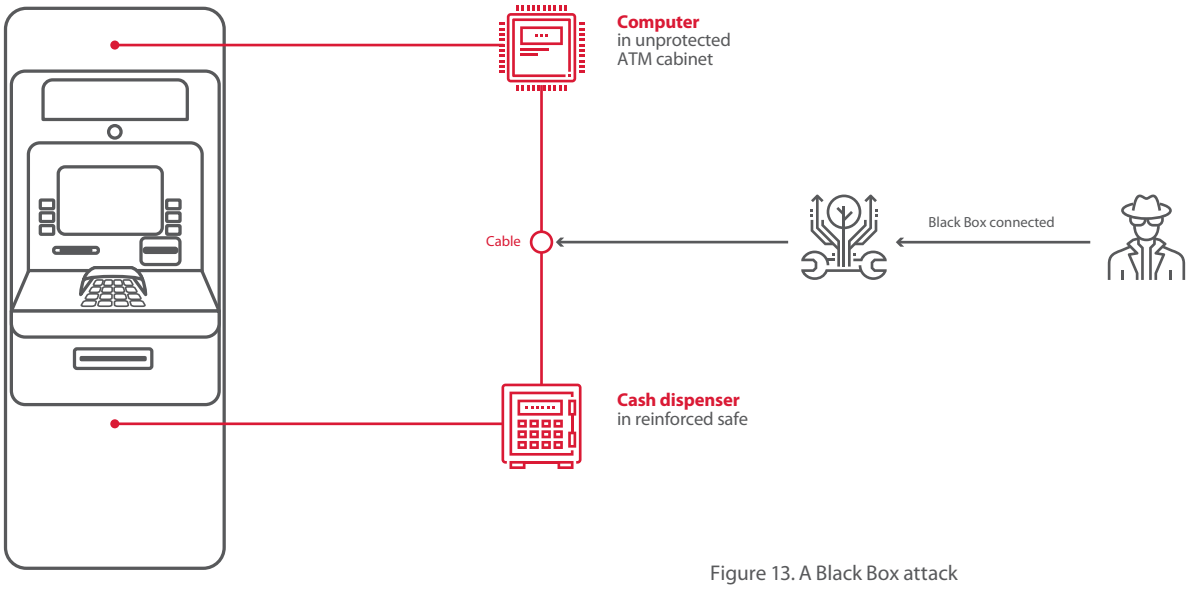


Figure 13. A Black Box attack

To prevent Black Box attacks, ATM vendors recommend using the latest XFS versions for strong encryption and physical authentication between the OS and dispenser. When physical authentication is present, encryption keys are sent only when legitimate access to the safe has been confirmed. However, inventive criminals have devised countermeasures of their own. For instance, in recent attacks in Mexico, one criminal group was able to emulate physical authentication with the help of an endoscope.

Encryption is not always well implemented, even in the latest software versions. For example, in 2018, Positive Technologies experts examining the APTRA XFS platform from NCR discovered vulnerabilities that made it possible to install a modified firmware version on the dispenser controller and therefore bypass physical authentication.

The vulnerable NCR protection system was used in half of studied ATMs. On 19 percent of ATMs, there were no protections against Black Box attacks at all.

Recommendations

1. Use physical authentication between the OS and dispenser to confirm legitimate access to the safe.
2. Encrypt data between the ATM OS and dispenser.
3. Use the latest versions of software and regularly install updates.
4. Log and monitor security events.
5. Consider using external devices (such as Cerber Lock or ATM Keeper) to protect against unauthorized connections to the cash dispenser.

Exit from kiosk mode

Vulnerable to attack	Access needed	Time needed
76% of ATMs	Physical access to ATM cabinet	**15** minutes

By design, an ordinary ATM user interacts with only one application, which displays information on the screen and processes input from the user. The application runs in kiosk mode, meaning that the user cannot run other programs or access OS functions in any way. By exiting kiosk mode, an attacker could bypass these restrictions and run commands in the ATM OS.

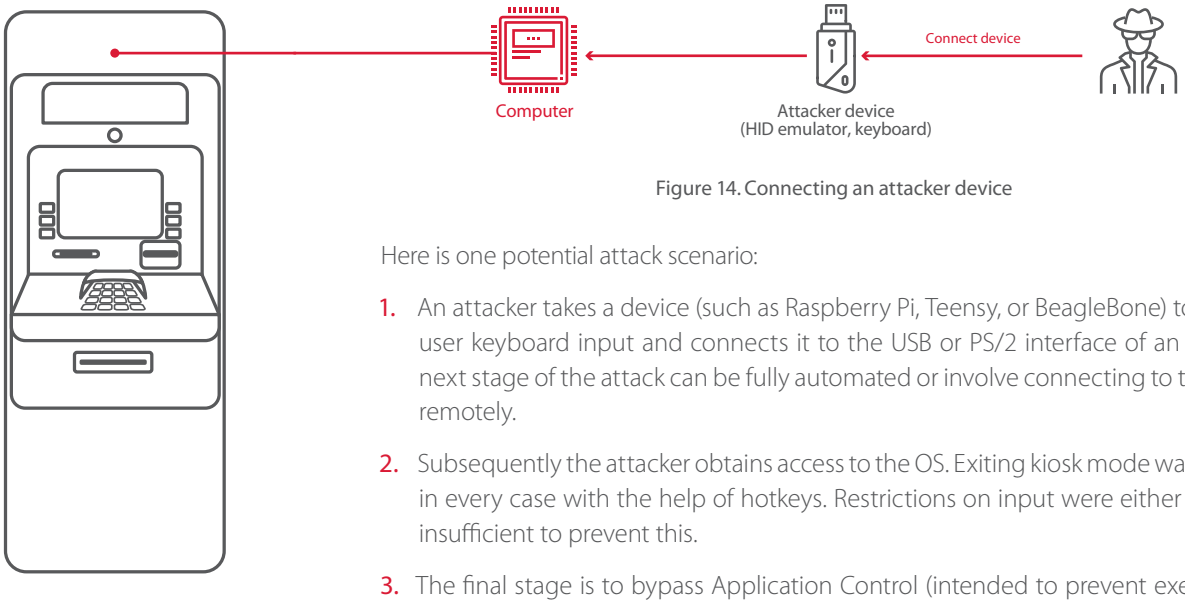


Figure 14. Connecting an attacker device

Here is one potential attack scenario:

1. An attacker takes a device (such as Raspberry Pi, Teensy, or BeagleBone) to emulate user keyboard input and connects it to the USB or PS/2 interface of an ATM. The next stage of the attack can be fully automated or involve connecting to the device remotely.
2. Subsequently the attacker obtains access to the OS. Exiting kiosk mode was possible in every case with the help of hotkeys. Restrictions on input were either absent or insufficient to prevent this.
3. The final stage is to bypass Application Control (intended to prevent execution of unwanted code) and gain the ability to send commands to the cash dispenser.

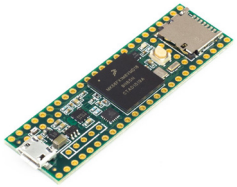


Figure 15. Teensy platform

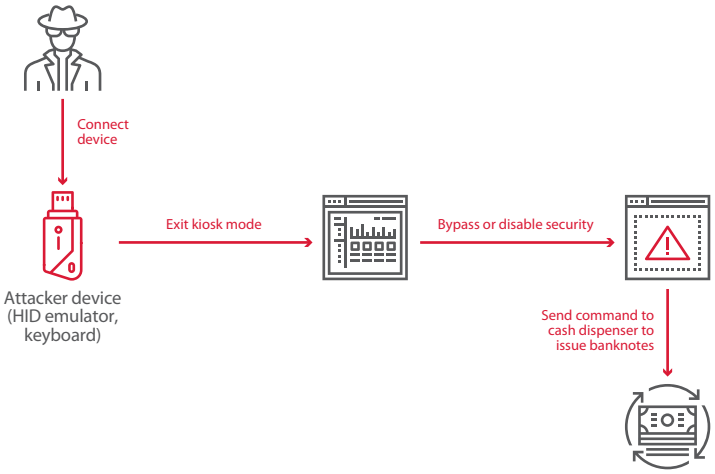


Figure 16. Exiting kiosk mode: attack scenario

Vulnerabilities found in testing

The tested ATMs contained configuration errors, primarily involving insufficient restriction of user account rights, as well as vulnerabilities in Application Control.

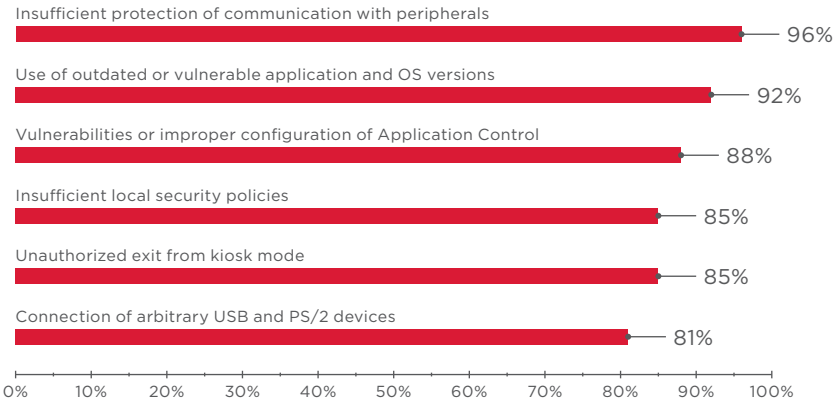


Figure 17. Incidence of vulnerabilities (percentage of tested ATMs affected)

Most tested ATMs allowed freely connecting USB and PS/2 devices. As a result, a criminal could connect a keyboard or other device imitating user input.

It is important to prevent entry of arbitrary information, such as certain key combinations that could be used to escape kiosk mode and obtain access to OS functions. Most tested ATMs ran special software to selectively disable key combinations. However, in 85 percent of cases, standard key combinations remained available, including Alt+F4 (close active window) and Win+Ctrl, Alt+Tab, and Alt+Shift+Tab (switch task). This technique allowed closing the window of the ATM kiosk application and disabling the applications responsible for blocking arbitrary keyboard input.

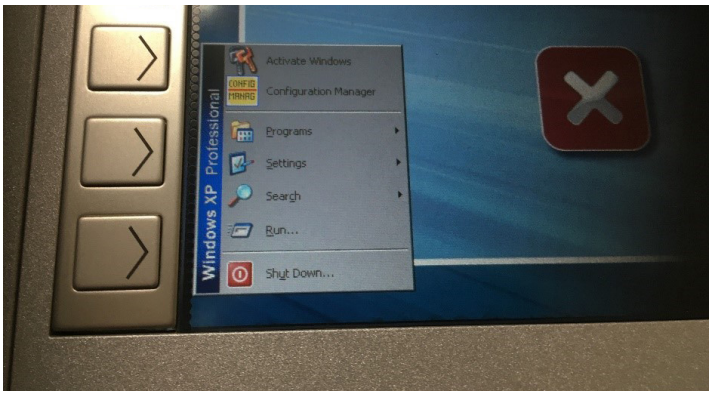


Figure 18. Exiting kiosk mode with keyboard shortcuts

Vulnerabilities for exiting kiosk mode may even be present in security software. For example, two ATMs ran software to record video and monitor security events. The application window was hidden, but it was found during testing that the window appears if the mouse cursor is placed in the corner of the screen. The application contained a function for editing files, which made it possible to access Windows Explorer, and subsequently any other software on the computer, such as Internet Explorer or FAR Manager.

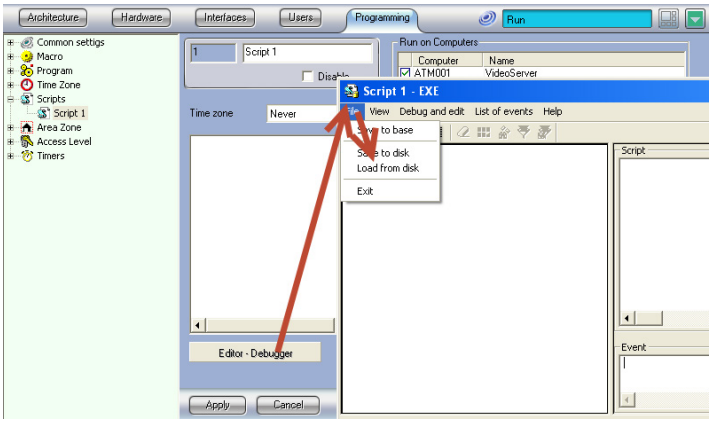


Figure 19. Exiting kiosk mode from Intellect software

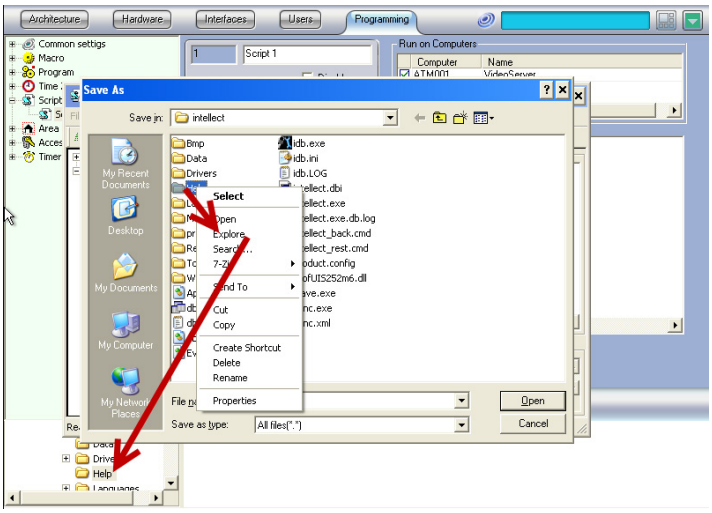


Figure 20. Exiting kiosk mode from Intellect software

Local security policies should be configured to deny users the ability to read/write files and launch arbitrary programs. On most tested ATMs, local security policies were poorly configured or absent entirely.

On 92 percent of tested ATMs, Application Control solutions were installed. They are designed to prevent execution of malicious code by allowing only whitelisted applications. The core weakness in Application Control configuration has to do with how the whitelist is created: any software already present during Application Control installation is classified as trusted, even if the software is not necessary for ATM functioning. Therefore, vulnerabilities in whitelisted software can be exploited to execute arbitrary code and disable protection. Vulnerabilities (some of them zero-day) were discovered in ATM security products as well.

Zero-day vulnerabilities

During their research, our experts have discovered zero-day vulnerabilities in Application Control products such as GMV Checker ATM Security, Kaspersky Embedded Systems Security, and McAfee Application Control (Solidcore). In 2018, Positive Technologies experts discovered three vulnerabilities in SafenSoft SoftControl: CVE-2018-13014, CVE-2018-13013, and CVE-2018-13012.

Vulnerability CVE-2018-13014 enables obtaining the password for accessing configuration parameters. The password was stored in cleartext in a database, which itself was in a folder available to ordinary users. An attacker could therefore change SafenSoft parameters and even disable protection entirely.

With the configuration password, an attacker could then exploit the second vulnerability, CVE-2018-13013. This vulnerability involves failure to correctly check the file msiexec.exe, which is used for software installation. An attacker can create a configuration in which signatures of .msi files are not checked and therefore enable launch of an arbitrary .msi file.

The third vulnerability, CVE-2018-13012, relates to the software update process. SafenSoft downloads a configuration file and update files via the insecure HTTP protocol. The integrity of these files is not checked, so an attacker could perform a man-in-the-middle attack to substitute the update files with malicious ones.

Recommendations

1. Use local OS policies or Device Control solutions to limit the ability to connect peripherals.
2. Disable standard key combinations that could be used to obtain access to OS functions.
3. Minimize user privileges as much as possible. Limit the ability to edit files, modify registry values, and run arbitrary programs.
4. Remove any software that is not necessary for ATM functioning. If removal is not possible, use security tools to restrict the software.
5. Double-check Application Control whitelists: when building a list of allowed applications, do not include unneeded built-in OS services or other applications that are not essential for ATM operation.
6. Enforce exclusive access to logical devices. Work with the vendor to implement API changes and authorization mechanisms.
7. Use the latest versions of software and regularly install updates.
8. Log and monitor security events.

Connection to hard drive

Vulnerable to attack	Access needed	Time needed
92% of ATMs	Physical access to ATM cabinet	**20** minutes

By connecting to the ATM hard drive, it is possible to bypass security and gain control of the cash dispenser. We will consider several related potential attack scenarios.

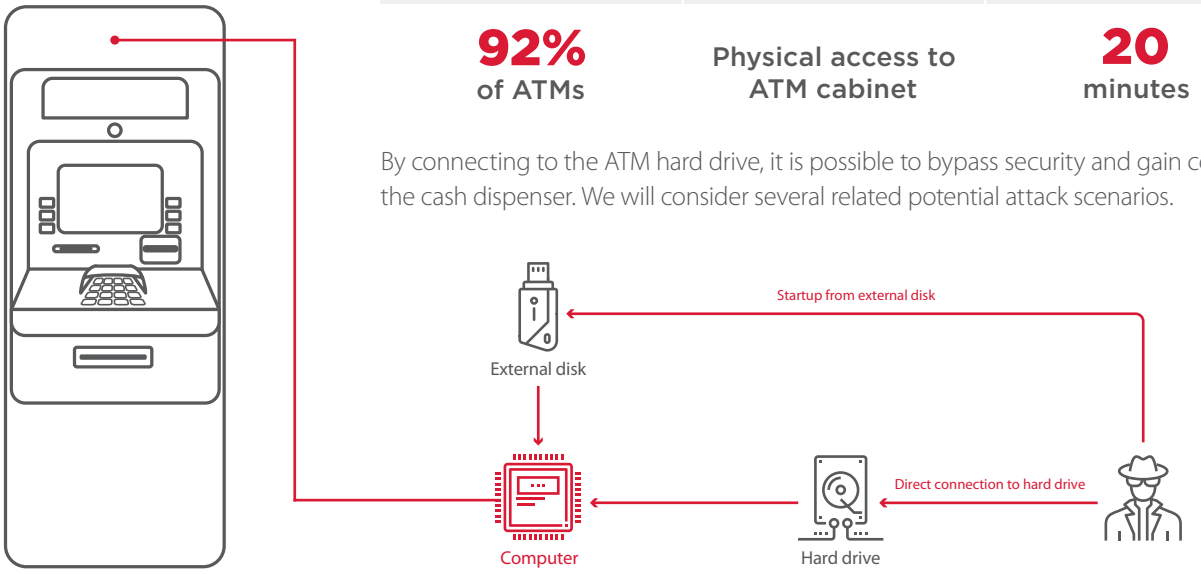


Figure 21. Connecting to the hard drive

Direct access to hard drive

At risk:
92% of tested ATMs

The simplest method is to connect directly to the hard drive. If the hard drive is not encrypted, the attacker can copy a malicious program with dispenser commands to it. Then the attacker needs to add this program to the Application Control whitelist by simply modifying the configuration files. When the ATM is subsequently restarted in ordinary ("secure") mode, the security software will launch and seemingly function—but the attacker can now run arbitrary code and malware. The attacker can even disable security software entirely, such as by deleting files from disk.

An attacker can also copy sensitive information from the hard drive (such as a particular application or even full image of the disk) and then use modified versions in future attacks.

Boot from external disk

At risk:
27% of tested ATMs

An attacker can start the ATM from an external disk in order to obtain access to the file system. The boot order is set in the BIOS, access to which should be password-protected. But on 23 percent of ATMs, the BIOS password was easy to guess. On 8 percent of ATMs, there was no password at all. In one case, it was not possible to obtain the administrator password. This would not have stopped an attacker though: no password was needed for ordinary user privileges and that user could change the boot order. In another case, testers could start up an ATM over the network with the help of Intel Boot Agent, overriding the BIOS boot order.

Having started the ATM from another disk, an attacker could connect the original hard drive and implement the scenarios already described in the case of direct connection to the ATM hard drive. The following screenshot demonstrates renaming of the McAfee Solidcore for APTRA driver on the ATM hard drive after the OS has been started from an external disk. As a result, McAfee Solidcore will not run when the ATM boots up from its internal hard drive.

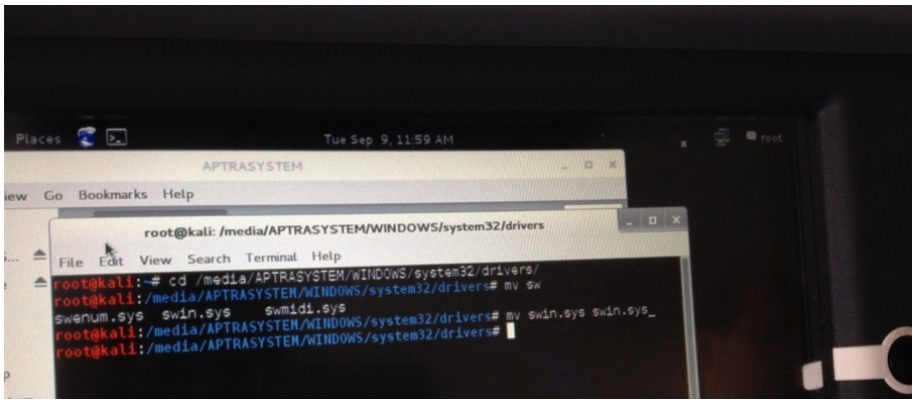


Figure 22. Renaming the McAfee Solidcore for APTRA driver

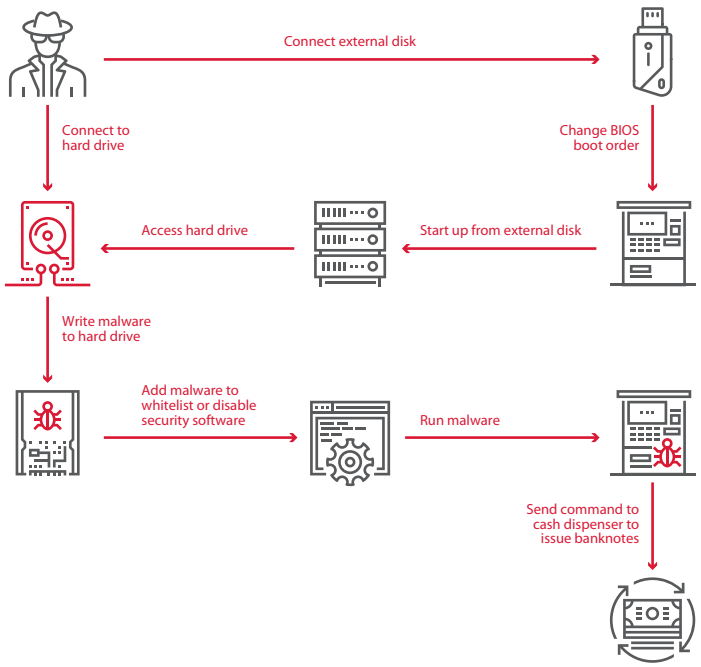


Figure 23. Connecting to a hard drive to write malware to it

Vulnerabilities found in testing

Vulnerabilities allowing access to the hard drive file system are caused by weaknesses in authentication for BIOS access and lack of disk encryption. Malware can communicate with the cash dispenser as the result of poor protection of peripherals, specifically a lack of authentication and encryption between the OS and devices.

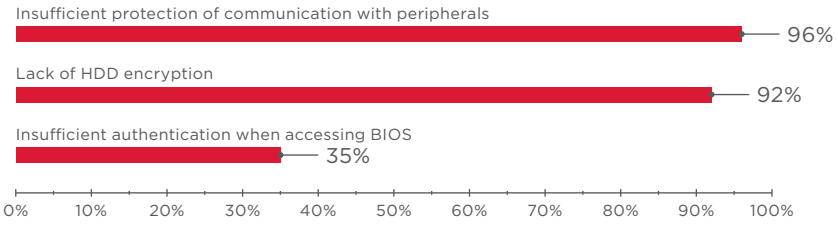


Figure 24. Incidence of vulnerabilities (percentage of tested ATMs affected)

The Ploutus malware family first surfaced in 2013. Attacks were initially concentrated in Latin America, but now (taking different variants of Ploutus into account) span the entire world. Total losses exceed $450 million.

Criminals use diverse methods to infect ATMs, including by writing malware directly to hard drives. The criminals remove the ATM hard drive, connect it to their own computer, plant a copy of the malware, and put the hard drive back in the ATM.

Recommendations

1. Encrypt ATM hard drives. Major vendor NCR has created guidelines for best encryption practices. These include transmitting encryption keys over the network, instead of storing them locally.
2. Enforce strict authentication for BIOS access.
3. Use UEFI instead of BIOS to ensure control of load memory integrity.
4. Allow startup only from the ATM hard drive. Forbid startup from external disks or over the network.

Boot mode modification

Vulnerable to attack	Access needed	Time needed
42% of ATMs	Physical access to ATM cabinet	**15** minutes

Starting the ATM operating system in a special mode can offer a way to bypass security. The tested ATMs had the following boot modes available:

- Kernel debug mode
- Directory Service Restore Mode
- Safe modes (Safe Mode, Safe Mode with Networking, Safe Mode with Command Prompt)

In these modes, some services and protection measures are disabled, creating an opportunity to exit kiosk mode. After starting the ATM in debug mode and connecting to the COM ports, an attacker can seize full control of the ATM by using the WinDbg utility.

Setting a different boot mode was possible on 88 percent of ATMs. In 42 percent of cases, the testers could develop this attack further and eventually withdraw cash.

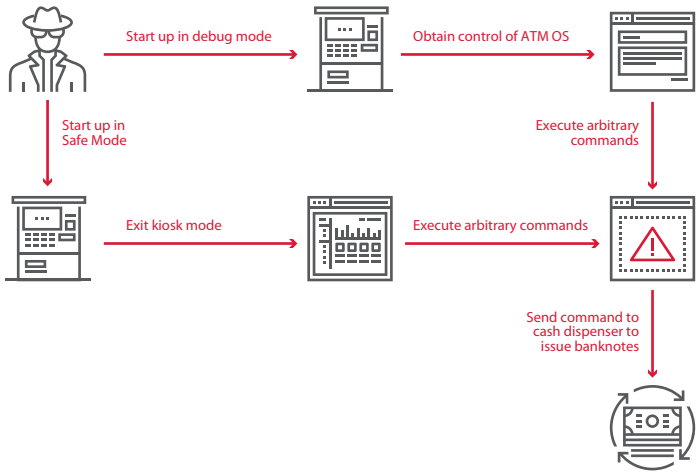


Figure 25. Changing the boot mode as part of a cashout attack

Recommendations

1. Disable the ability to select boot mode from the Windows loader.
2. Disable access to debug mode via COM/USB interfaces and over the network.

Card data theft

Vulnerable to attack	Access needed	Time needed
100% of ATMs	Physical access to ATM cabinet or Access to ATM network	**15** minutes

The magnetic stripe of bank cards contains information needed to perform transactions. Although the magnetic stripe can fit up to three tracks, usually only two (Track1 and Track2) are used. Track1 contains the card number, expiration date, service code, and owner name. It may also contain the PIN Verification Key Indicator, PIN Verification Value, and Card Verification Value. Track2 duplicates all the information on Track1 other than the owner name.

Paying with a magnetic stripe at a POS terminal or withdrawing cash from an ATM requires only reading Track2. So attackers seek to copy the information from Track2. This information can be used to create fake card duplicates, which are offered for sale on the darkweb. So-called card dumps account for a quarter of all information sold on the darkweb. The average cost of a single card is $9.

For years, criminals placed physical shims (skimmers) on a card reader in order to read information directly from the magnetic stripe. Banks caught on and now widely implement measures to thwart skimming. Nonetheless, data can still be stolen even without skimmers. Interception is possible at two stages:

- During data transmission between the ATM and processing center
- During data transmission between the ATM operating system and card reader

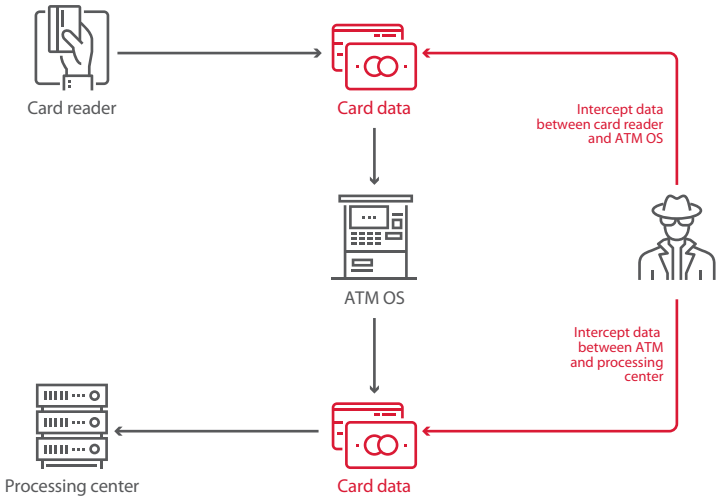


Figure 26. Attacks aimed at card data theft

We will briefly consider some of these attack scenarios. They have much in common with the scenarios discussed already and take advantage of the failure to perform data encryption and authentication at critical stages of the transaction process.

Interception of data between ATM and processing center

At risk:
58% of tested ATMs

This attack is possible because the full value of Track2 is sent in cleartext and no encryption is applied to traffic between the ATM and processing center at the application level (nearly all ATMs use the NDC and DDC protocols, which do not employ encryption). So by connecting to the ATM network and listening to network traffic, an attacker can obtain information about bank cards.

Wireshark · Follow TCP Stream (tcp.stream eq 0)

```
00000000  00                                                .
00000001  01 60 31 31 1c 30 30 32  1c 1c 1c 31 3a 1c 3b 34  .`11.002 ...1:.;4
00000011  38 [redacted]                                    3d  8[redacted]=
00000021  31 38 30 38 32 30 31 32  34 38 31 33 30 30 30 30  18082012 48130000
00000031  30 35 33 30 3f 1c 1c 41  41 20 20 20 20 20 20 1c  0530?..A A      .
```

Figure 27. Intercepting Track2 in cleartext

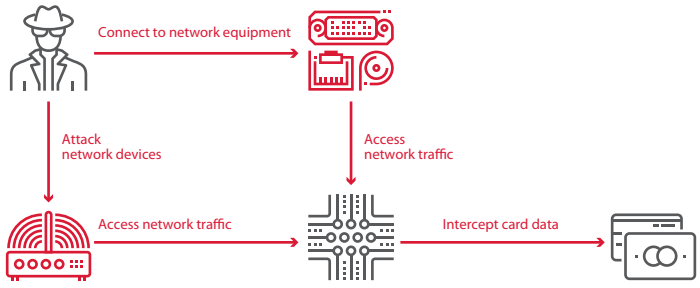


Figure 28. Intercepting data between the ATM and processing center

Interception of data between OS and card reader (via USB or COM port)

At risk:
100% of tested ATMs

A special device is placed between the ATM computer and card reader in order to intercept the contents of the magnetic stripe of bank cards. These attacks are possible because communications with the card reader are not authenticated or encrypted; card data is sent in cleartext. Such deficiencies were found on all tested ATMs.

Interception of data between OS and card reader (with malware)

At risk:
100% of tested ATMs

Malicious hardware is not necessary for reading data from the card reader if the attacker is able to install malware on the ATM. This can be accomplished in any of the ways described in this report: changing the boot mode or starting up from an external disk, connecting directly to the hard drive, attaching a device to emulate user input, or performing a network attack.

None of the ATMs performed authentication when exchanging data with the card reader. Therefore, any device could access it. All an attacker would need to do is run arbitrary code in the ATM OS.

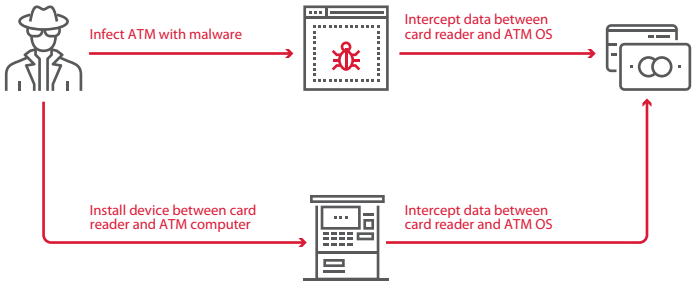


Figure 29. Intercepting data between the card reader and ATM OS

Skimer, the ATM malware known since 2009, continues to be developed. A new version of Skimer discovered in 2016 can steal data from bank cards, including PIN codes. Criminals installed the malware via the bank's internal network or physical access to the ATM. An infected ATM could accumulate data for months on end without arousing suspicion. Then the criminals collected their haul. A criminal or accomplice would walk up to the ATM, insert a special card, and enter a session key to activate the malware. Then Skimer could write all data to the card or print it on receipt paper. Besides Skimer, other malware for stealing bank card information includes Ripper and Suceful.

In 2016, criminals in Japan with cloned cards made off with $12.7 million in just three hours. In August 2018, a similar attack hit Cosmos Bank in India: criminals pilfered over $11 million with the help of cloned cards.

Recommendations

1. Encrypt data exchange with the card reader. Do not send the full contents of Track2 in cleartext.
2. Implement the recommendations given in this report to prevent arbitrary code execution.
3. Implement the recommendations given in this report to prevent network attacks that target traffic between the ATM and processing center.

Conclusion

Logic attacks on ATMs are growing in popularity, with losses running in the millions of dollars. Although ATM owners bear the brunt of the threat, bank clients may be victimized as well—especially in the case of card cloning attacks. In our security analysis work, we continually uncover vulnerabilities related to network security, improper configuration, and poor protection of peripherals. Taken together, these flaws provide criminals with the ability to steal ATM cash or obtain card information. More often than not, security mechanisms are a mere nuisance for attackers: our testers found ways to bypass protection in almost every case. Since banks tend to use the same configuration on large numbers of ATMs, a successful attack on a single ATM can be easily replicated at greater scale.

The recommendations in this report are intended to harden ATMs against logic attacks. As the difficulty of exploitation rises, the likelihood of crime decreases. To reduce the risk of attack, the first step is to physically secure the ATM cabinet and surroundings. Exploiting most of the vulnerabilities we found would be impossible without access to the on-board computer and peripheral ports. Another key step is to log and monitor security events, for quickly reacting to threats as they arise. Regular security analysis of ATMs is important for timely detection and remediation of vulnerabilities. Security analysis may also include reverse engineering of ATM software, such as Application Control, XFS-related software, and network equipment firmware. Such testing offers uniquely powerful results due to identification of zero-day vulnerabilities and subsequent measures to protect against novel attack vectors.

ptsecurity.com
info@ptsecurity.com

ATM-Vulnerabilities-2018_A4.ENG.0004.03