Abstract
Although the incident response has always been a crucial component of information security, security administrators frequently ignore it. Whereas, Reverse engineering focuses on the difficult issue of analyzing legacy software code in the absence of appropriate documentation. This paper proposes an approach to understanding cyber security Incident Response and the services it provides followed by Reverse Engineering resources and the practical analysis of a malware named “Alice ATM Malware” in detail.

Key words: Cybersecurity Incident, Indicator of compromise, Digital defenses, Computer Security Incident Response Team, IDA de-compilers, Reverse Engineering.

1. Introduction
Highly skilled attackers provide a continuing threat to organizations. Attackers constantly discover new ways to breach organizations’ digital defenses to steal information or destroy their operations. The threat landscape is rapidly changing (Ahmad et al. 2019). There is a lot of ongoing research on how to reinforce these digital defenses, but relatively little is done to improve the process that takes over when things go wrong: incident response (IR) [1].

When digital defenses fail, cybersecurity incident response teams are at the forefront and must intervene to reinstitute services and catch up to the capabilities of current crackers. Security professionals must find new ways to breach systems to ensure that they are always prepared for the next threat. The authors offer an analysis of a modern-day malware attack on ATM machines, which comprises a chain of events that was not possible in previous years. The authors will describe the process and its limitations. The authors will also detail the practical analysis of malware ‘Alice ATM Malware’.

Problems are impeding the incident response team’s capacity to respond to cyber-attacks (Nyre-Yu et al. 2019) [2]. By removing these obstacles, organizations may be better able to respond to incidents in general. One method to achieve this is through training, particularly employing training scenarios, which can boost team cybersecurity performance by developing skills and spotting possible flaws. The most significant information technology advancements are right in front of us right now. Given that approximately one billion computers in our world are currently connected to the Internet and that mobile telephony services and e-commerce have converged, enormous amounts of information can go from one network to another with only
one request, command, or click (Global Reach, 2005). To this degree, numerous technologies, platforms, and infrastructures are flourishing to give services to the end user, who now serves as the target point because it is the user who asks for services, accesses networks, and resources, and needs security and privacy [3].

Expertise in fields like forensic investigation and malware reverse engineering is frequently needed for incident response. Reverse engineering is the process through which a variety of items, including software, types of machinery, and architectural structures, are disassembled to obtain design data and is also known as back engineering. The reverse engineering method typically involves disassembling the parts of larger, more important products. The reader will learn about reverse engineering's common principles, applications, stages, and future in this article. It demonstrates how Reverse Engineering is constantly developing and influencing the idea of cyber security [4].

Software development can also benefit from reverse engineering since it allows developers to examine their code and identify potential flaws that were overlooked during software development but that an adversary could find through reverse engineering. In Cybersecurity it is important because it enables the extraction of Indicators of Compromise (IoC) from samples [5]. Typically IoC is file’s hashes, registry keys, import function and export functions, the programming language used, compilation date, IPs, emails, and even text strings that are present in the code, which are the traces left by attackers [6] [7].

2. Incident Response

The handling of diverse security incidents, cyber threats, and data breaches involves an organized technique called incident response. A cyber attack or live incident's cost is to be identified, contained, and reduced using incident response techniques [8]. Although it is not the end-all solution, a solid incident response (IR) plan can seal a potential weakness to avoid more attacks. The response is a part of incident handling, which in turn looks at the coordination, logistics, and planning required to deal with a problem. This kind of work is normally handled by the Computer Security Incident Response Team (CSIRT), with assistance from the Security Operation Center. While incident management is the primary function of CSIRT, it also has reporting, analysis, and reaction responsibilities. Before these phases, the incident must be located and promptly reported [9]. The function of a SOC Analyst becomes crucial at this point.

2.1 Incident Response Services

The most efficient incident response is carried out rapidly by trained responders. Organizations frequently lack the funding necessary to keep a fully functional incident response team on duty around the clock [10]. Working with an outside organization that provides qualified incident response services is one alternative.

Getting involved with these organizations offers the following advantages:

2.1.1 Availability

The cost and impact of an attack on the organization are reduced the faster the incident response team gets to work. Cybersecurity issues can happen at any time, and getting in touch with incident response team members
after hours might be challenging. To improve coverage and availability, professional incident response companies have numerous staff teams on hand.

2.1.2 Experience
Managing security issues improperly can increase costs and harm a company. For instance, a ransomware attack might cause a system to become unstable, making it unlikely that encrypted data will be restored after a system restart. Professional incident responders have the experience necessary to accurately and efficiently address such security issues.

2.1.3 Specialized expertise
Reverse engineering malware and forensic analysis are two skills that are frequently used in incident response. Even though the majority of businesses won't require these talents in-house, a specialized incident response team has access to the professionals they require to successfully manage cybersecurity problems.

2.1.4 Controlling all aspects of the incident response procedure
All of the organization's incident response requirements should be met by outsourced incident response providers. This includes putting incident response plans in place, controlling identified intrusions, and thwarting potential attacks.

3. Incident Response Plan Phases

It is a set of guidelines that must be followed during each stage of incident response. The components of a good incident response plan include a clear communication strategy, directives defining the duties and responsibilities of each person and organization, and protocols that must be followed at all times [11].

3.1 Preparation
The steps a business should take in the case of a disruptive incident are outlined in an effective incident response strategy. The plan starts by describing how a company should reduce the danger of a data leak [12]. Organizational data protection policies should be in line with security objectives and technology defenses throughout the preparation stage. You must, at the very least, guarantee that staff members have received training on information security. They should ideally also have specialized training in incident response. To make sure your sensitive data is adequately protected, you should audit your systems as well.

3.2 Identification
The second component of incident response planning deals with the measures an organization takes to ascertain when one of its systems has been compromised [13]. You are better able to stop you may quickly recognize an incursion from an assault. You can save time and money even if it isn't possible by limiting the damage and hastening the response effort.

The following inquiries should be addressed when determining a security incident:

- Who found the opening?
- How much of a breach is there?
- Had an impact on our operations?
- Where did the compromise originate?

3.3 Containment
The third phase discusses the steps you should
take to minimize harm after being infiltrated. Depending on the circumstances, this can require taking action to remove the criminal hacker from your networks or to isolate the already compromised data [14].

During this stage, you should consider whether systems need to be shut down or removed as well as whether there are any rapid fixes for vulnerabilities.

3.4 Eradication
Fixing the fault that caused the data breach is the aim of phase four of a cyber incident response strategy. Again, the specifics will depend on the type of incident, but right now you need to determine how the information was disclosed and how to get rid of the threat [15].

For instance, you would remove the malicious software and isolate the affected areas if your firm had been compromised by malware. In the meantime, you would lock down a worker's account if the attack resulted from their login information being stolen by a malicious hacker.

3.5 Recovery
Getting your systems back online is the penultimate step in responding to a cyber incident once the threat has been eradicated. In some circumstances, this will be trickier to do than in others, but it's an important step that needs to be taken seriously. Without a strong recovery process, you might still be vulnerable to attacks, which would make the injury worse. As part of the recovery process, you should test and monitor the impacted systems after the issue has been fixed. This guarantees that the measures you implement the function as planned and offers you the chance to fix any errors [16].

3.6 Lessons Learned
The final phase in the cyber incident response strategy is reviewing the occurrence and identifying potential areas for improvement. Your incident response team needs to meet to go over the parts of the plan that worked and any problems you encountered.

It is important to review the process at every stage. Discuss what happened, why it happened, what you tried to stop it, and what may have been done differently. For instance, was the documentation effective and clear, and did the plan include any gaps?

Before having this conversation, one to two weeks should have passed after the security incident; this will allow everyone to think about the incident in retrospect while still keeping it fresh in everyone's memory [17].

Instead of berating team members for prior mistakes, this stage's objective is to avoid inefficiencies from occurring again. Failures in the processed signal that the documentation was either unclear, the proper course of action wasn't specified, or staff training wasn't adequate.

4 Reverse engineering
The procedure of obtaining knowledge or designing blueprints from everything created by humans is known as reverse engineering. The idea has likely existed since the Industrial
Revolution, long before computers or other contemporary technology. It closely resembles scientific research, in which an investigator seeks to identify the "blueprint" of an atom or the human mind. Reverse engineering is different from a traditional scientific study in that the artifact under investigation is human-made, as opposed to a natural event in scientific research [18]. When such information is lacking, reverse engineering is typically used to fill in the gaps in knowledge, ideas, and design philosophy.

4.1 Resources Used in Reverse Engineering
Multiple tools are used to perform reverse engineering. These tools can help to debug, decompile and disassemble the application.

4.1.1 Debuggers
GDB is a debugger for programming that may also be used to decipher binary code. While the assembly code is running, you can view the information in memory and registers [19]. Additionally, breakpoints can be added anywhere in the application using debuggers.

4.1.2 Disassembler
Machine code is converted into a human-readable format using a disassembler. Because disassembled code lacks programmers' comments and annotations, reading it is more challenging than reading source code.

4.1.3 De compilers
Although IDA is difficult to use and needs extensive programming knowledge, its technical level accurately captures the fundamental nature of reverse engineering (Seo et al., 2019). DE-compiled is the process of translating a compiled program into a higher-level symbolic language that humans can comprehend, and it particularly makes use of reverse engineering methods.

4.2 Stages in Reverse engineering
By creating models that describe the current program and the assumed goal, reverse engineering can be accomplished. Three main phases make up this process:

- Recovering from implementation. Prepare a preliminary model while learning about the application quickly.
- Adaptive design. Foreign key references should be resolved and the database's mechanics reversed.
- Retrieval of analysis. Eliminate any inaccuracies in the model and design artifacts.

4.2.1 Implementation Recovery
You prepare an early model for reverse engineering during implementation recovery. The first model should only reflect the implementation and contain no inferences because it will be used as a reference.

Reading through the most recent documentation and getting acquainted with a program is the first step. The resulting context makes it simpler to interact with application professionals and clarifies the developer's intention. This project ought to be finished in a few hours. Even though what you learn is unrelated to the actual reverse engineering, it is crucial since it enables you to make more
accurate observations as you go.

The database structure is then typed in manually or automatically into a modeling program. Some tools have can read an RDBMS's system tables and seed a model. If you utilize these tools, you ought to at the very least glance over the database architecture to get a sense of the development approach.

4.2.2 Design Recovery
You undo the database's mechanics during design recovery and just carry out simple operations. Conjecture and interpretation should wait until the analysis-recovery stage. In most cases, design recovery may be carried out independently, without assistance from application expertise. You fix three main problems at this stage.

Identity: For the prospective entity type keys, unique indexes will typically be defined. Otherwise, search for uncommon data combinations, which can point to a candidate key but not confirm it. You can also guess potential keys by looking at names and styling conventions. A suspected foreign key may imply a comparable candidate key.

Foreign keys: The most difficult aspect of design recovery is often identifying foreign keys or references between different tables. Foreign keys may be indicated by names and data types that match. Foreign keys and their referents can be declared by developers in some DBMSs, such as RDBMSs, but the majority of legacy applications do not use this feature.

Queries: When they are available, queries can be used to improve your understanding of identity and foreign keys.

Design reclamtion may result in optimizations and flaws, but it still reflects the DBMS paradigm in its ultimate form. The model will rarely be finished in practice. Some of the structure's components might be unclear.

4.2.3 Analysis Recovery
The last step is analysis and recovery. The model is interpreted, improved, and made more abstract. During this stage is when you should speak with any available application professionals. Recovery from analysis consists of four basic steps.

Clarification: Eliminate any existing design artifacts. For instance, file and database access keys do not include any essential information and are purely design options, thus they are not required to be included in an analytical model.

Redundancy: Remove derived data type if it improves database design or if it was included for the wrong reasons. You might need to examine the data to determine whether a data structure is a duplicate.

Errors: Resolve any leftover database problems. This phase of analysis recovery is necessary since you need to completely understand the database before you can say that the developer committed a mistake. An apparent error in the early stages might have been a fair procedure or the result of inadequate database understanding [20].

Integration of a model. Various information sources can result in various models. For instance, structure and data analysis is typically used to develop a reverse-engineered
model. A forward-engineered model could be created using a user manual. The final analysis model must incorporate all independent models.

5. Practical Demonstration

The following shown is the 32-bit executable file. The Alice ATM malware was discovered for the first time in November 2016 as part of an ATM malware research study with Europol EC3, however, researchers believe it has been present since 2014.

![Executable file](image1)

**Figure 1:** Executable file

This malware is detected on Virus total and 54 security vendors and 1 sandbox flagged this file as malicious on Virus Total.

![Results of Virus Total](image2)

**Figure 2:** Results of Virus Total

These are the libraries that are used by this malware.

<table>
<thead>
<tr>
<th>library</th>
<th>blacklist</th>
<th>type</th>
<th>functions</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntdll.dll</td>
<td>-</td>
<td>implicit</td>
<td>6</td>
<td>NT Layer DLL</td>
</tr>
<tr>
<td>user32.dll</td>
<td>-</td>
<td>implicit</td>
<td>12</td>
<td>Multi-User Windows USER API Client DLL</td>
</tr>
<tr>
<td>kernel32.dll</td>
<td>-</td>
<td>implicit</td>
<td>7</td>
<td>Windows NT BASE API Client DLL</td>
</tr>
<tr>
<td>comctl32.dll</td>
<td>-</td>
<td>implicit</td>
<td>1</td>
<td>Common Controls Library</td>
</tr>
<tr>
<td>msxfs.dll</td>
<td>x</td>
<td>implicit</td>
<td>10</td>
<td>Extension for Financial Services (XFS)</td>
</tr>
</tbody>
</table>

**Figure 3:** Libraries used by the malware
At this point, it detects that is it running on an

Figure 8: When digital defenses fail, cybersecurity

al. 2019). There is a lot of ongoing research on

specialized peripherals like ATMs, XFS offers

to be the first XFS API function a program

connected to the Internet and that mobile

advancements are right in front of us right

the traces left by attackers [6] [7].

variety of items, including software, types of

one request, command, or click (Global Reach,

managed message box that appears when we

organization are reduced the faster the incident

The cost and impact of an attack on the

The most efficient incident response is carried

instance, a ransomware attack might cause a

Had an impact on our operations?

•

•

better able to stop you may quickly recognize

organization takes to ascertain when one of its

needs to be taken seriously. Without a strong

some circumstances, this will be trickier to do

hacker.

meantime, you would lock down a worker's

software and isolate the affected areas if your

For instance, you would remove the malicious

The procedure of obtaining knowledge or

mistakes, this stage's objective is to avoid

any problems you encountered.

The final phase in the cyber incident response

was the documentation effective and clear, and

identifying potential areas for improvement.

The second component of incident response

starts by describing how a company should

in-house, a specialized incident response team

Project Alice

EVENT and USER EVENT to receive data for

contemporary technology. It closely resembles

GDB is a debugger for programming that may

4.1.1  Debuggers

The first model should only reflect the

engineering during implementation recovery.

You prepare an early model for reverse

inaccuracies in the model and design

were the documentation effective and clear, and

identifying potential areas for improvement.

The final phase in the cyber incident response

errors [16].

The following shown is the 32-bit executable

command for execution.

this function.

specified by the service to a service provider.

apparent error in the early stages might have

design recovery and just carry out simple

design recovery is often identifying foreign

unique indexes will typically be defined.

should wait until the analysis-recovery stage.

program. Some tools have can read an

model will rarely be finished in practice. Some

Eliminate any existing design

Ntdll.dll:

A module containing NT system functionality

is called ntdll.dll. The NT kernel functions are

<table>
<thead>
<tr>
<th>functions (36)</th>
<th>blacklist (3)</th>
<th>ordinal (0)</th>
<th>library (5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RtlCaptureStackBackTrace</td>
<td>x</td>
<td>-</td>
<td>ntdll.dll</td>
</tr>
<tr>
<td>RtlMoveMemory</td>
<td>x</td>
<td>-</td>
<td>ntdll.dll</td>
</tr>
<tr>
<td>VerSetConditionMask</td>
<td>x</td>
<td>-</td>
<td>ntdll.dll</td>
</tr>
<tr>
<td>RtlUnwind</td>
<td>-</td>
<td>-</td>
<td>ntdll.dll</td>
</tr>
<tr>
<td>RtlZeroMemory</td>
<td>-</td>
<td>-</td>
<td>ntdll.dll</td>
</tr>
<tr>
<td>RtlFillMemory</td>
<td>-</td>
<td>-</td>
<td>ntdll.dll</td>
</tr>
</tbody>
</table>

Figure 4: Functions that are used from this library.

Comctl.dll:

A module called Comctl32.dll houses standard

GUI elements used by Windows programs.

| InitCommonControls | - | - | comctl32.dll |

Msxfs.dll:

Microsoft didn’t provide much information on

| WFSCleanUp | - | - | msxfs.dll |
| WFSOpen | - | - | msxfs.dll |
| WFSGetInfo | - | - | msxfs.dll |
| WFSExecute | - | - | msxfs.dll |
| WFSLock | - | - | msxfs.dll |
| WFSRegister | - | - | msxfs.dll |
| WFSFreeResult | - | - | msxfs.dll |
| WFSUnlock | - | - | msxfs.dll |
| WFSClose | - | - | msxfs.dll |
| WFSStartUp | - | - | msxfs.dll |
Malicious Strings:

<table>
<thead>
<tr>
<th>encoding (2)</th>
<th>size (bytes)</th>
<th>location</th>
<th>blacklist (3)</th>
<th>hint (31)</th>
<th>value (190)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ascii</td>
<td>24</td>
<td>0x00013C2</td>
<td></td>
<td>function</td>
<td>RHCaptureStackTrace</td>
</tr>
<tr>
<td>ascii</td>
<td>13</td>
<td>0x00013DE</td>
<td></td>
<td>function</td>
<td>RHFillMemory</td>
</tr>
<tr>
<td>ascii</td>
<td>13</td>
<td>0x00013EE</td>
<td>x</td>
<td>function</td>
<td>RHMoveMemory</td>
</tr>
<tr>
<td>ascii</td>
<td>9</td>
<td>0x00013FE</td>
<td></td>
<td>function</td>
<td>RHFUnwind</td>
</tr>
<tr>
<td>ascii</td>
<td>13</td>
<td>0x0001404</td>
<td></td>
<td>function</td>
<td>RHFZerOMemory</td>
</tr>
<tr>
<td>ascii</td>
<td>19</td>
<td>0x000141A</td>
<td>x</td>
<td>function</td>
<td>VarSetConditionMask</td>
</tr>
<tr>
<td>ascii</td>
<td>13</td>
<td>0x0001446</td>
<td></td>
<td>function</td>
<td>AnimateWindow</td>
</tr>
<tr>
<td>ascii</td>
<td>9</td>
<td>0x000145E</td>
<td></td>
<td>function</td>
<td>EmiDialog</td>
</tr>
<tr>
<td>ascii</td>
<td>10</td>
<td>0x0001474</td>
<td></td>
<td>function</td>
<td>GetDlgItem</td>
</tr>
<tr>
<td>ascii</td>
<td>8</td>
<td>0x000148A</td>
<td></td>
<td>function</td>
<td>ILHand</td>
</tr>
<tr>
<td>ascii</td>
<td>8</td>
<td>0x000148B</td>
<td></td>
<td>function</td>
<td>SetFocus</td>
</tr>
<tr>
<td>ascii</td>
<td>11</td>
<td>0x0001526</td>
<td></td>
<td>function</td>
<td>ExitProcess</td>
</tr>
<tr>
<td>ascii</td>
<td>13</td>
<td>0x0001528</td>
<td></td>
<td>function</td>
<td>SetInfoWritePtr</td>
</tr>
<tr>
<td>ascii</td>
<td>10</td>
<td>0x000153A</td>
<td></td>
<td>function</td>
<td>LocalAlloc</td>
</tr>
<tr>
<td>ascii</td>
<td>9</td>
<td>0x0001544</td>
<td></td>
<td>function</td>
<td>LocalFree</td>
</tr>
<tr>
<td>ascii</td>
<td>9</td>
<td>0x0001550</td>
<td></td>
<td>function</td>
<td>LocalOne</td>
</tr>
<tr>
<td>ascii</td>
<td>18</td>
<td>0x0001576</td>
<td></td>
<td>function</td>
<td>InitCommonControls</td>
</tr>
<tr>
<td>ascii</td>
<td>10</td>
<td>0x000159A</td>
<td></td>
<td>function</td>
<td>WFSSetStartup</td>
</tr>
<tr>
<td>ascii</td>
<td>8</td>
<td>0x00015A8</td>
<td></td>
<td>function</td>
<td>WFSStop</td>
</tr>
<tr>
<td>ascii</td>
<td>9</td>
<td>0x00015B4</td>
<td></td>
<td>function</td>
<td>WFSUnlock</td>
</tr>
<tr>
<td>ascii</td>
<td>13</td>
<td>0x00015C0</td>
<td></td>
<td>function</td>
<td>WFSSetResult</td>
</tr>
<tr>
<td>ascii</td>
<td>11</td>
<td>0x00015D0</td>
<td></td>
<td>function</td>
<td>WFSRegister</td>
</tr>
<tr>
<td>ascii</td>
<td>10</td>
<td>0x00015E3</td>
<td></td>
<td>function</td>
<td>WFSExec</td>
</tr>
<tr>
<td>ascii</td>
<td>10</td>
<td>0x00015F5</td>
<td></td>
<td>function</td>
<td>WFSGetInfo</td>
</tr>
<tr>
<td>ascii</td>
<td>10</td>
<td>0x000160E</td>
<td></td>
<td>function</td>
<td>WFSCleanUp</td>
</tr>
</tbody>
</table>

Figure 5: Malicious Strings

Dynamic Analysis:

When running this malware it asks for the pin and it only takes 123 as a pin. It didn’t take any other number. When entering any other number it wasn’t proceeding and when you entered 123 as pin it automatically processes the next step [21].

The "operator panel" is opened by entering a special PIN that is of 3 digits based on the terminal ID of the ATM.
Maybe here it defines the password.

```assembly
.text:00401031 ; int (__stdcall *dword_401031)(_DWORD)
.text:00401031 dword_401031 dd 0 ; DATA XREF: DialogFunc+165+r
.text:00401031 ; sub_401AE9+36+r
.text:00401035 ; align 4
.text:00401035 db 0
.text:00401039 dword_401039 dd 0 ; DATA XREF: DialogFunc+170+r
.text:00401040 ; align 10h
.text:00401040 db 0 .text:00401041 ; CHAR a123[4]
.text:00401041 a123 db '123',0 ; DATA XREF: sub_401AE9+D54o
.text:00401044 ; CHAR string2[2]
.text:00401044 String2 db '0',0 ; DATA XREF: DialogFunc+13C+o
.text:00401045 ; sub_401AE9:loc_401BF6+0

lea ebx, [ebp+1Param]
push ebx ; lParam
push 104h ; wParam
push 0Dh ; Msg
push 7D5h ; nIDDlgItem
push [ebp+hd1g] ; hd1g
call SendDlgItemMessageA
push offset a123 ; "123"
push ebx ; lpString1
call lstrlenA
or eax, eax
jnz short loc_401BF6
```

After entering the pin it asked to input the cassette id. The loaded cassettes holding the money are visible when the "operator panel" is opened. The values ID, Bills count, Bill value, Currency, and Result are shown. It displays all cassettes containing money in the machine.

Figure 6: Running the malware

Figure 7: Password Defined
A connection is made between an application function. In this function it calls sub_401047. In the second call, it calls the sub_401078 function. In this function it calls WFSStartUp, a connection is made between an application and the XFS Manager by WFSStartUp. It has to be the first XFS API function a program calls. XFS functions cannot be supplied by an application until a successful WFSStartUp has finished.

Figure 8: Entering the PIN

At this point, it detects that is it running on an ATM. When running on XFS (Extension for Financial Services)-based machines, it accepts input. For Microsoft Windows-based financial applications, particularly those that use specialized peripherals like ATMs, XFS offers a client-server architecture.

Figure 9: Second call

In the second call, it calls the sub_401078 function. In this function it calls WFSStartUp, a connection is made between an application and the XFS Manager by WFSStartUp. It has to be the first XFS API function a program calls. XFS functions cannot be supplied by an application until a successful WFSStartUp has finished.

After that, it made calls to the
DialogBoxParamA

```assembly
xor edi, edi
push edi ; lpModuleName
call GetModuleHandleA
mov hInstance, eax
push 3
call sub_401047
push edi ; dwInitParam
push offset sub_401AE9 ; lpDialogFunc
push edi ; hWndParent
push 7D0h ; lpTemplateName
push hInstance ; hInstance
call DialogBoxParamA
call sub_401078
push edi ; uExitCode
call ExitProcess
start endp
```

**Figure 10:** DialogBoxParamA

This function takes five parameters. It takes one function as a parameter.

```c
void __noretum start()
{
    hInstance = GetModuleHandleA(0);
    sub_401047(3);
    DialogBoxParamA(hInstance, (LPCSTR)0x7D0, 0, sub_401AE9, 0);
    sub_401078();
    ExitProcess(0);
}
```

**Figure 11:** Function as parameter

This function that is going as a parameter is managed message box that appears when we run the program and it takes decisions based on provided data is correct or not.
The highlighted function is responsible to connect the dispenser1 of the ATM.
Alice connects to ATM’s CurrencyDispenser1 peripheral, and no other hardware; therefore, the criminal does not need to issue any command via PIN pad.
WFSClose is used to end a session or a series of service requests between the application and the designated service that was started using WFSOpen.

Then it calls WFSFreeResult, which informs the XFS manager that a memory buffer that was dynamically allocated by a service provider is ready to be released. An application uses this function to deallocate the memory.

A second call was placed. All messages of the defined classes are forwarded to the window indicated in the hWndReg argument by WFSRegister, which enables event monitoring for the specified service via the specified window. For instance, the application can call WFSRegister with the parameters SYSTEM EVENT and USER EVENT to receive data for both system and user events.
At this point, it decides to see if input caste is available or not if available it calls the highlighted function if not then it moves on else part and displays an error.

**WFSLock** Establishes the application’s sole authority over the designated service. Before beginning the transaction, the application needs to make sure that it has access to all the devices and that no other program will be able to utilize them until the transaction is finished.

```c
int __stdcall sub_40111F(int a1, int a2, int a3)
{

    int v4; // [esp+Ch] [ebp-8h] BYREF
    int v5; // [esp+10h] [ebp-4h]

    v5 = 0;
    WFSLock(a1, 0, &v4);
    if (v4)
        WFSSetResult(v4);
    v4 = 0;
    WFSExecute(a1, a2, a3, 0, &v4);
    v5 = v4;
    WFSUnlock(a1);
    return v5;
}
```

So here before starting the transaction, the application ensures that it has access to all devices and until a transaction is made, no other program can use them. For this, it uses **WFSLock & WFSUnlock**. After calling the function WFSLock it will call the next command for execution.

Utilizing the WFSLock function and its companion WFSUnlock allows for this.

**WFSEexecute** communicates a command specified by the service to a service provider. To run commands supplied by the service, use this function.

A service that has been locked by a previous WFSLOCK FUNCTION is released by WFSUnlock.
Examining the next function, it monitors if the selected cassette is available or not. It gives an error if the input cassette is unavailable.

```assembly
.text:00401260 .CHAR text[]
.text:00401260 text db 'Selected cassette is unavailable!',0
.text:00401260 ; DATA XREF: sub_401B7412740
.text:00401283 ; CHAR aCanTDispenseRe[]
.text:00401283 aCanTDispenseRe db 'Can',27h,'t dispense requested amount. Error %d occurred!',0
.text:00401283 ; DATA XREF: sub_401B74F81c
.text:00401287
```

Before exiting the process it calls the function.

In this function, it calls the WFSCleanUp function. An application is unplugged from the XFS manager via WFSCleanUp.

**The Flow of XFS APIs used:**

---

**Key Points:**

This executable file, when provided is not named when I start reverse engineering I found a titled ‘Project Alice’. Alice ATM first asks for a pin after entering a pin opening the operator panel reveals the loaded cassettes that hold the money. While malware has one main function that it uses to connect to the currency dispenser peripheral in the ATM. During the reverse engineering process, I didn’t see that if it attempts to connect to other ATM hardware such as a PIN pad. So one thing is clear at this point it is not controlled by commands issued via Pin pad.

**How exactly it works:**

1. It first calls WFSStartUp API to connect the application with the XFS manager.

2. After that it calls WFSRegister to enable event monitoring for the specified service.
3. Next, it makes a call to the WFSLock API to make sure that the application has access to all the devices before beginning the transaction and that no other application can use them until the transaction is finished. Utilizing the WFSLock function and its companion WFSUnlock allows for this.

4. Next, a call to WFSFreeResult alerts the XFS manager that a dynamically allocated memory buffer from a service provider has to be freed. An application uses this function to deallocate the memory.

5. Next, it calls WFSEexecute to transmit a command specific to the service to a service provider. To run commands supplied by the service, use this function. Then it calls WFSUnlock to release a service that has been locked by a previous WFSLOCK function.

6. To disconnect the application from the XFS manager, it makes a final call to WFSCleanUp.

6. Conclusion

The globe has suddenly become a global village as a result of the extraordinary rise of information technology. As it happens, it has made the world smaller and knowledge flow more freely. Additionally, it has increased internet vulnerabilities, threats, scams, and criminal activity. The privacy of people, organizations, and states has been violated by the accessibility, user-friendly hacking tools, and complexity of cyberattacks. A good computer and network security life cycle, which comprises countermeasures, detection, and reaction, now includes incident response as a crucial component. An organization's information security policy should contain the necessary provisions, and from there, planning and organization are essential for a successful incident response effort. The planning and organizing for the incident response also includes developing a suitable incident response architecture, planning resource requirements, planning the use of technology, developing incident response procedures, cooperating with other teams and organizations, and developing appropriate metrics. Moreover, Reverse engineering encompasses a wide range of tasks, such as system data analysis and the DE compilation and disassembly of executable files and libraries. Reverse engineering is a technique used in computer security to analyze malware activities and develop solutions to stop them.

7. References


