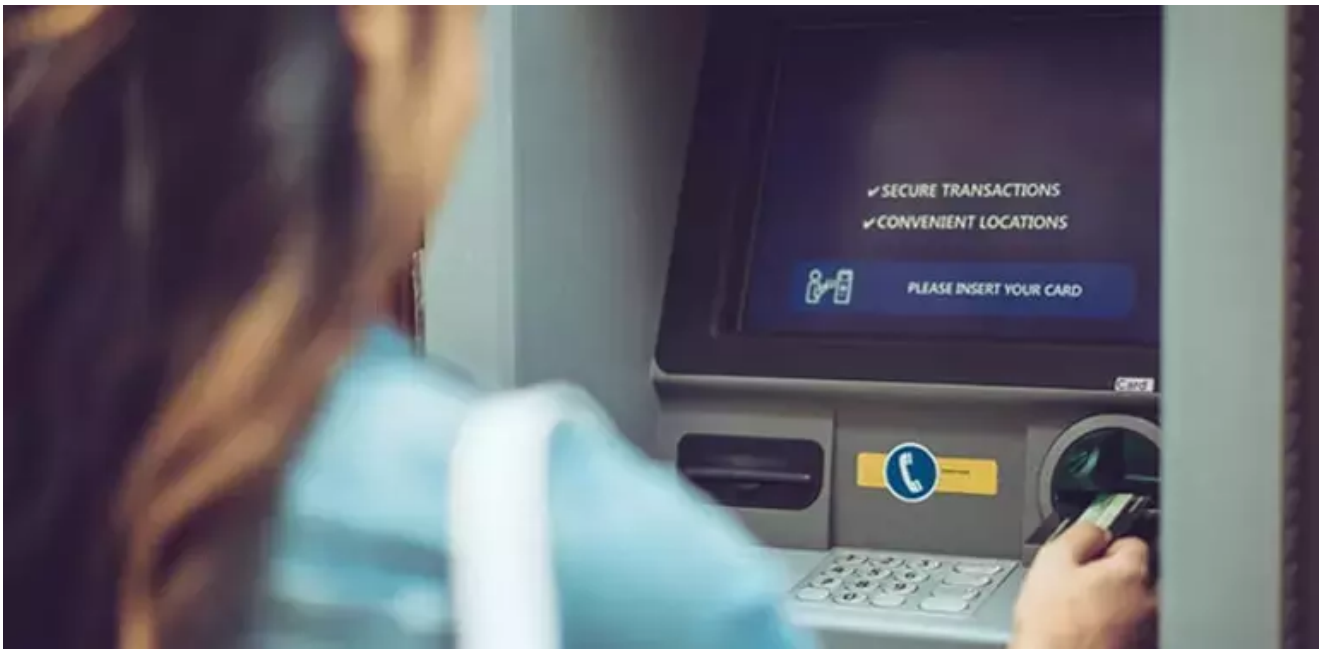
Meet GreenDispenser: A New Breed of ATM Malware

proofpoint.com/us/threat-insight/post/Meet-GreenDispenser

September 22, 2015



Share with your network!

September 24, 2015 Thoufique Haq

By Thoufique Haq

On the heels of recent disclosures of ATM malware such as Suceful [1], Plotus [2] and Padpin [3] (aka Tyupkin), Proofpoint research has discovered yet another variant of ATM malware, which we have dubbed GreenDispenser.

GreenDispenser provides an attacker the ability to walk up to an infected ATM and drain its cash vault. When installed, GreenDispenser may display an 'out of service' message on the ATM -- but attackers who enter the correct pin codes can then drain the ATM's cash vault and erase GreenDispenser using a deep delete process, leaving little if any trace of how the ATM was robbed.

Deployment and Operation

Initial malware installation likely requires physical access to the ATM, raising questions of compromised physical security or personnel. Once installed, GreenDispenser is similar in functionality to Padpin but does exhibit some unique functionality, such as date limited operation and a form of two-factor authentication.

Specifically, GreenDispenser like its predecessors interacts with the XFS middleware [4], which is widely adopted by various ATM vendors. The XFS middleware allows software to interact with the peripherals connected to the ATM such as the pinpad and the cash dispenser by referencing the specific peripheral name. GreenDispenser has the ability to target ATM hardware from multiple vendors using the XFS standard. It achieves this by querying for peripheral names from the registry hive before defaulting to hardcoded peripheral names.

The malware strains Proofpoint inspected were coded to run only if the year was 2015 and the month was earlier than September, suggesting that GreenDispenser was employed in a limited operation and designed to deactivate itself to avoid detection. Furthermore, GreenDispenser employs authentication using a static hardcoded PIN, followed by a second layer of authentication using a dynamic PIN, which is unique for each run of the malware. The attacker derives this second PIN from a QR code displayed on the screen of the infected ATM. We suspect that the attacker has an application that can run on a mobile phone with functionality to scan the barcode and derive the second PIN -- a two-factor authentication of sorts. This feature ensures that only an authorized individual has the ability to perform the heist. In addition, GreenDispenser has the capability to perform a deep delete after the heist to prevent forensic analysis and IR investigations.

Technical Details

An initial inspection of the IAT (Import Address Table) in GreenDispenser shows usage of various XFS APIs through msxfs.dll in order to interface with the XFS middleware.

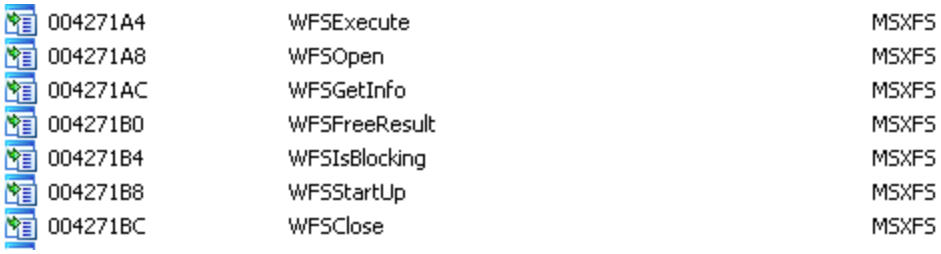


Figure 1: IAT shows usage of XFS APIs

Once run, GreenDispenser performs a check to verify that the current year is 2015 and the current month is earlier than September. If these conditions are not met, then GreenDispenser simply quits.

```
push    eax             ; lpSystemTime
movq    [esp+9Ch+var_96], xmm0
call    ds:GetSystemTime
mov     eax, 7DFh       ; Year = 2015
cmp     [esp+98h+var_98], ax
jnz     loc_40C62F
cmp     word ptr [esp+98h+var_96], 9 ; Month < Sept
jnb     loc_40C62F
```

Figure 2: Time bound checks

If the checks pass, GreenDispenser proceeds to create a mutex called “dispenserprgm” to ensure that only a single instance of GreenDispenser is running. It then creates a second desktop environment on the ATM called “dDispW” and creates a window in the second desktop called “Dispenser”. This window is created using the window style “WS_EX_TOPMOST“ to ensure that it overlays all other windows on the ATM screen. GreenDispenser may initially display a message on the screen indicating that the ATM is out of service as shown in Figure 3. It is interesting to note that while this instance displays a message in English (or somewhat close to it), other instances displayed an out order message in Spanish with the string “Temporalmente fuera de servicio ”.

Figure 3: Fake out of service message

GreenDispenser then initiates a session to the XFS manager using WFSStartUp and attempts to query the registry location “HKEY_USERS\ .DEFAULT\XFS\LOGICAL_SERVICES\class=PIN” to obtain the peripheral name for the Pinpad [5]. If not found it defaults to “Pinpad1” which is the pinpad peripheral name on

specific ATMs. GreenDispenser then waits in an infinite loop for input from the pinpad. It accepts input from the pinpad using a call to WFSExecute with the command set to “WFS_CMD_PIN_GET_DATA” as shown in Figure 4.

```
push    eax
push    0
lea     eax, [ebp+var_48]
push    eax
movzx   eax, word ptr [esi]
push    198h            ; WFS_CMD_PIN_GET_DATA
push    eax
call    ds:WFSExecute
```

Figure 4: API call to accept input from pinpad

If the right static Pin is provided it then displays the screen shown in Figure 5 prompting for a second Pin.

Figure 5: Screen after entering hardcoded static PIN

The contents of the QR code are randomly seeded and subjected to encryption using the Microsoft CryptoAPI followed by Base64 encoding, but we have chosen to forgo further discussion of details in order to avoid potential misuse of infected ATMs. We suspect that the attacker has an application that can run on a mobile phone with functionality to scan the barcode and derive the second PIN. Once the attacker enters the correct secondary PIN into the pinpad a second menu is shown (Figure 6), which allows access to the cash dispenser.

Figure 6: Screen after entering dynamically generated PIN

If the dispense cash option is selected, GreenDispenser attempts to query the registry location “HKEY_USERS\ .DEFAULT\XFS\LOGICAL_SERVICES\class=CDM” to find the peripheral name for the cash dispenser. If not found, it defaults to “CurrencyDispener1” which is the cash dispenser peripheral name on specific ATMs. It then makes a call to WFSExecute with the command set to “WFS_CMD_CDM_DISPENSE” and a timeout of 12000 to dispense cash.

```
push    eax
push    1D4C0h          ; 12000
lea     eax, [ebp+var_3C]
push    eax
movzx   eax, word ptr [ebx]
push    12Eh            ; WFS_CMD_CDM_DISPENSE
push    eax
mov     [ebp+var_1F], 0
mov     [ebp+var_23], edi
mov     [ebp+var_36], 1
call    ds:WFSExecute
```

Figure 7: API call to dispense cash

GreenDispenser also has the ability to delete itself, as may be seen in the options offered in the malware interaction menu. Typically when a file is deleted, the operating system removes the reference pointer to the data but not the data itself. This allows files to be recovered using disk editors and forensics tools later in time. To prevent this forensics analysis GreenDispenser performs a deep delete using sdelete to remove itself from the ATM. The sdelete executable is imbedded within GreenDispenser, which is written to disk as “del.exe” and run with the batch script shown in Figure 7. Again, such an action would presumably exist to deter forensic and IR investigations after the heist.

```
:start
tasklist /FI "IMAGENAME eq <malware.exe>" 2>NUL | find /I /N "<malware.exe>">NUL
if "%ERRORLEVEL%"=="0" goto start
"<path>\del.exe" /accepteula -p 3 -q "<path>\<malware.exe>"
del "<path>\del.exe"
del "<path>\<malware.exe>"
shutdown -t 0 -r -f
del "%~f0"
```

Figure 8: Batch script using sdelete to perform a deep delete

Conclusion

ATM malware continues to evolve, with the addition of stealthier features and the ability to target ATM hardware from multiple vendors. While current attacks have been limited to certain geographical regions such as Mexico, it is only a matter a time before these techniques are abused across the globe. We believe we are seeing the dawn of a new criminal industry targeting ATMs with only more to come. In order to stay ahead of attackers financial entities should reexamine existing legacy security layers and consider deploying modern security measures to thwart these threats.

References

[1] https://www.fireeye.com/blog/threat-research/2015/09/suceful_next_genera.html

[2] http://www.symantec.com/connect/blogs/texting-atms-cash-shows-cybercriminals-increasing-sophistication

[3] https://securelist.com/blog/research/66988/tyupkin-manipulating-atm-machines-with-malware/

[4] https://en.wikipedia.org/wiki/CEN/XFS

[5] https://doc.axxonsoft.com/confluence/display/atm70en/Configuring+the+connection+to+the+card+reader+service+provider

IOCs

Hashes(SHA256)

20a1490b666f8c75c47b682cf10a48b7b0278068cb260b14d8d0584ee6c006a5

50db1f5e9692f217f356a592e413e6c9cb31105a94efc70a5ca1c2c73d95d572

7544e7a798b791cb36caaa1860974f33d30bc4659ceab3063d1ab4fd71c8c7e0

77850f738ba42fd9da299b2282314709ad8dc93623b318b116bfc25c5280c541

b7e61f65e147885ec1fe6a787b62d9ee82d1f34f1c9ba8068d3570adca87c54f

Mutex:

dispenserprgm

Created desktop name:

dDispW

Created window name:

Dispenser

Registry queries:

HKEY_USERS\ .DEFAULT\XFS\LOGICAL_SERVICES\class=PIN

HKEY_USERS\ .DEFAULT\XFS\LOGICAL_SERVICES\class=CDM