Operation SyncTrek

By S2W

Published: 2021-02-17 · Archived: 2026-04-06 00:44:41 UTC

19 min read

Feb 15, 2021

Deep Analysis of TinyPos from the Clop Gang's Crime Scene | **S2WLAB Talon**

Author:

(Sojun Ryu), S2WLAB Talon

> This report is published in collaboration with Theori.

Introduction

Clop ransomware is well-known enterprise-targeted ransomware that has been active since early 2019. Clop ransomware is an ongoing threat actively attacking the world. TA505, known as the operator of Clop ransomware, has been targeting the financial sector since 2014. Many experts and reports from them have asserted that there is a strong correlation between Clop ransomware and the TA505 threat actor.

By posting 13 victims as a starter, Clop ransomware has launched a leak site titled “CL0P^_- LEAKS” around March 2020. One of the main purposes of running the leak site is to threat victim companies with their stolen data containing sensitive and confidential information which will be exposed to the public (Dark Web) if ransom negotiation fails.

We have observed that TA505 used TinyPoS malware while performing the Clop ransomware attack during the recent incident response. TinyPoS is the malware discovered in 2015 when PoS malware was on its prevalence. This malware is a ‘Memory Scraper’ that targets PoS (Point-of-Sale) or ATM in order to steal Track 1 or Track 2 data from the process memory. TinyPoS exfiltrates stolen data through the network or saves the data as a file, and the latter was used in this incident. The adversary continuously attempted to collect the hijacked data after storing it at the main collection server located inside the victim’s company.

While investigating past cases related to TinyPoS, we confirmed additional connections with not only Clop ransomware but DoppelPaymer and ProLock Ransomware. We have evidenced that TinyPoS has been deployed from the servers that were used by DoppelPaymer. Moreover, there is a high similarity between TinyPoS and ProLock when decoding the binary before execution.

The above examples show that cybercriminals are cooperating somehow and that adversaries targeting the financial sector in the past are continuing to attempt to steal card data. Besides, TinyPoS steadily appears in PoS-related incidents according to the report published by VISA. The report emphasizes that there are still threat actors attempting to steal card data and it is necessary to review and rebuild our defense line to ensure that we are well prepared for such attacks. We hope you find this report to be helpful when researching these malware and threat groups.

Background Knowledge

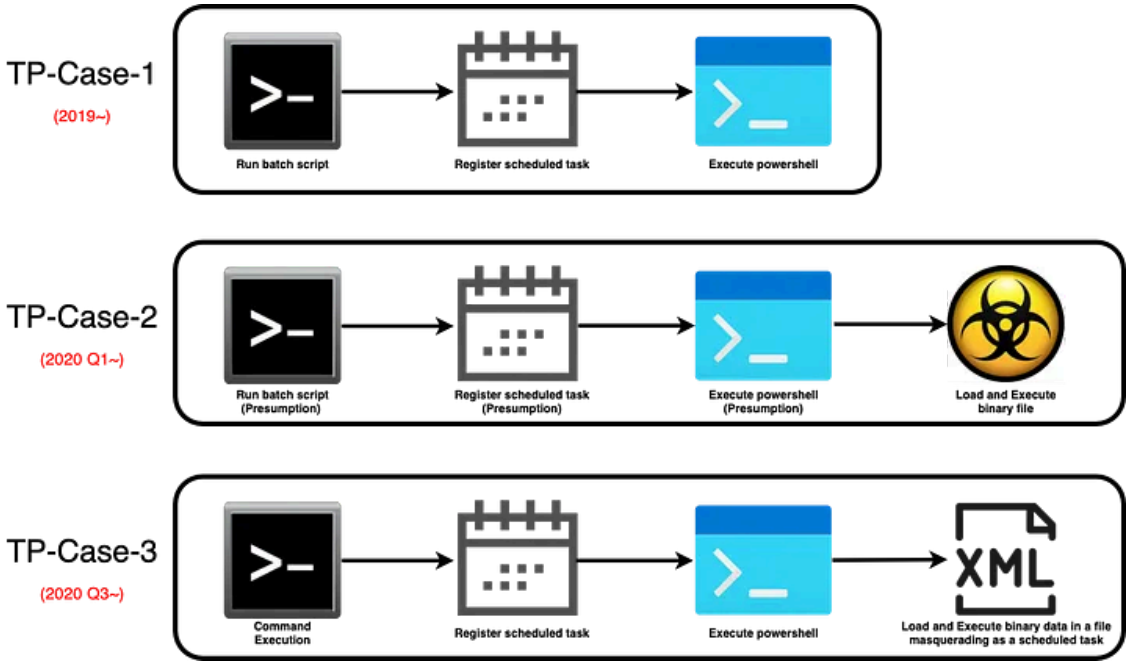
The card data that PoS malware attempts to steal usually represents Track 1 or Track 2. It is stored in the Magnetic Stripe on the back of the card. It contains important data such as card number, expiration date, and owner’s name.

The card data that PoS malware attempts to steal usually represents Track 1 or Track 2. It contains important data such as card number, expiration date, and owner’s name. Track 1 includes the card number, owner name, expiration date, etc. Track 2 is almost identical to Track 1 data, but the owner’s name is not included. The majority of credit card payment systems utilize Track 2 data since it only contains the necessary information for transactions as well as authentication. Upon leakage, Track 2 data is known to be the most popular data traded at Deep/Darkweb because it can be utilized to create fake cards.

Summary

TinyPoS is a very small size within 8KB and operates inside memory as assembly or binary form, so it needs a launcher and loader to execute it. The adversary used various methods for this, and each method is as follows.

Press enter or click to view image in full size



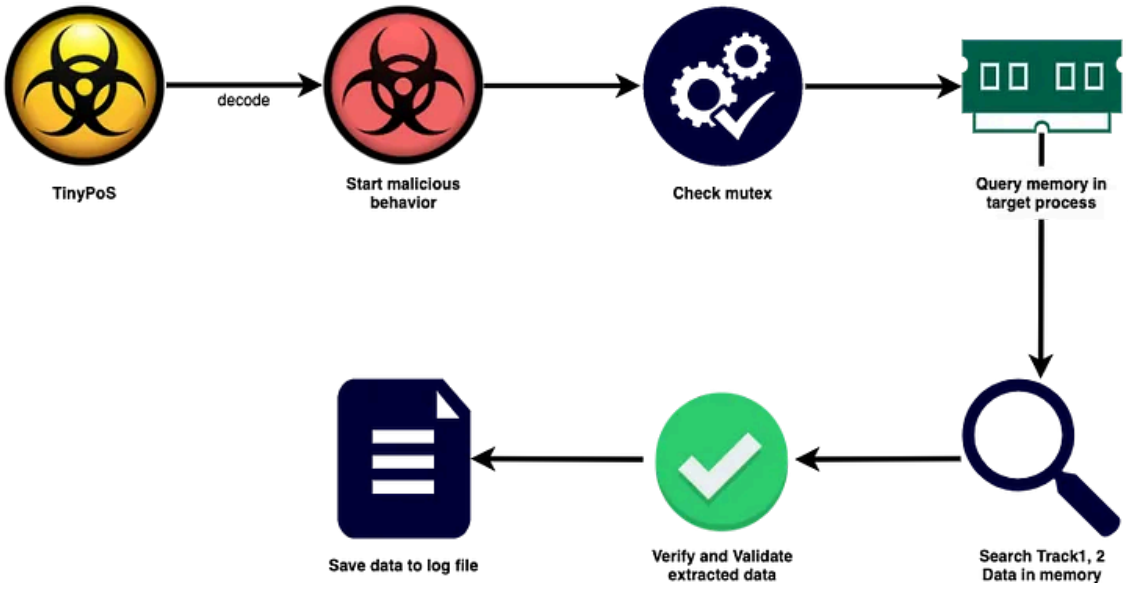
[Figure 2] Classification of TinyPos Installation

In the case of TinyPoS, it steals data from memory in real-time. The adversary periodically executes the malicious code using the scheduler. It has been confirmed that the scheduler registration is mainly executed by the command of the previously inserted remote control malicious code or through a batch file, and the launcher that runs TinyPoS is composed of Powershell.

TP-Case-1 created a PowerShell script file to run TinyPoS, and TP-Case-2 saved TinyPoS binary data as a file. In TP-Case-3, TinyPoS was hidden in a file disguised as a regular file. The adversary created all three cases by disguised as a commonly used file name or a file name similar to the file being used by the compromised server.

The executed TinyPoS can read the memory of the specific process that the adversary commanded or access every memory of the process that is not stated in the exclusion list in order to search for Track 1 and Track 2. After verifying the expiration date of the extracted data, the expired data is excluded. After that, only data verified by the Luhn algorithm is finally taken.

Press enter or click to view image in full size

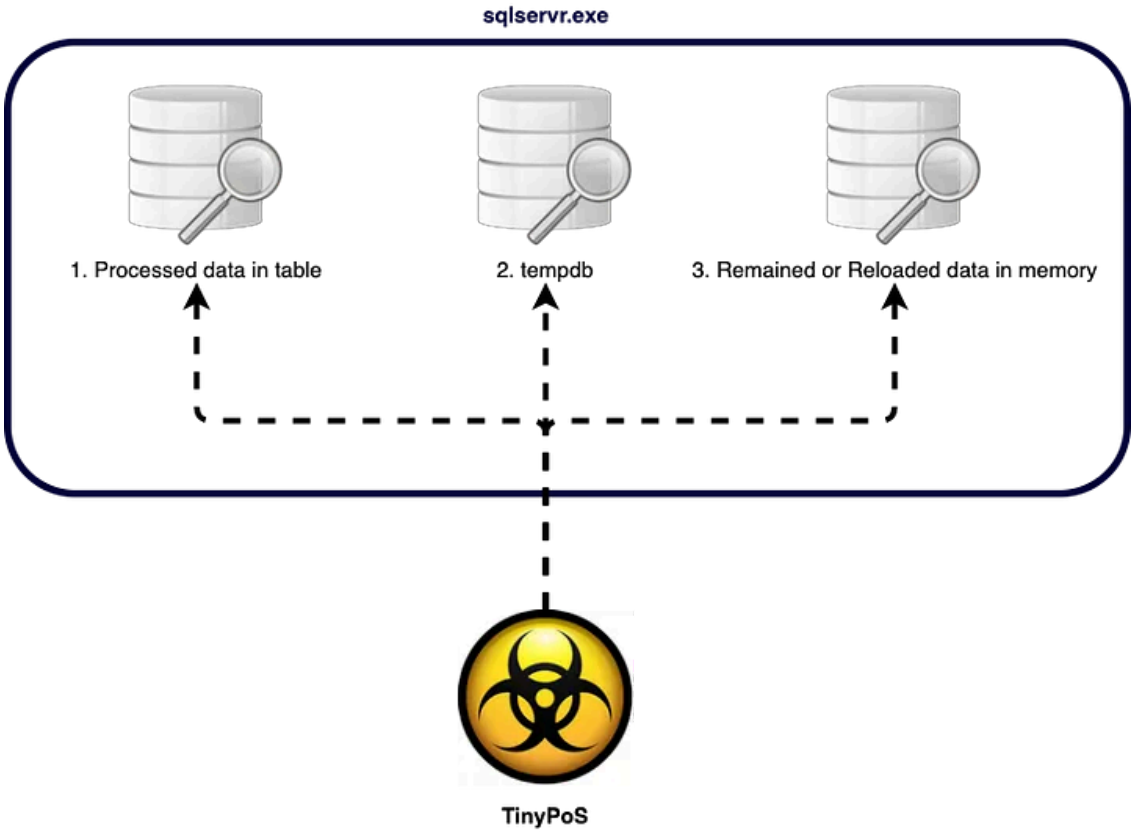


[Figure 3] Overview of TinyPoS's Code Flow

We have evidenced some of TinyPoS techniques used by the adversary attempting to target only SQL Server process (sqlservr.exe) for the purpose of stealing data. sqlservr.exe is a well-known process for managing MS-SQL. In this sense, it can be regarded as the adversary's intention to steal card data from MS-SQL. The Track 1 and 2 stored in the SQL Server database can be stolen from the memory in the following three cases.

1. When Track 1 and 2 data stored in the DB table
2. When Track 1 and 2 data found to be remained in "tempdb" due to the fact that such data processed via DB procedure, etc.
3. When the deleted Track 1 and 2 data are reloaded into memory through table lookup ⇒ It can be possible only when the metadata of Track data has been removed by deletion query

Press enter or click to view image in full size



[Figure 4] How to steal card data from Database

Analysis of TinyPos

The adversary installed and removed various types of TinyPoS malware in the victim server. While staying at the victim server, the task name and log file path were continuously changed and installed. This allowed us to ingest files that the adversary did not accidentally delete.

There were 3 cases of TinyPoS installation, but it is possible that there were actually more versions. It was confirmed that all cases were registered and operated in the scheduler through batch scripts or commands.

Analysis and comparison of TinyPoS installation methods

[TP-CASE-1]
TP-CASE-1 TinyPoS was included in PowerShell scripts, and the adversary used it around a year.

Batch Script of TP-Case-1

Tasks registered with a batch file execute a PowerShell script file containing TinyPoS every 6 hours. In the log file generated by TinyPoS, the detected Track 1 and 2 data and the detected process name are recorded line by line. It is initially presumed that the adversary performed the lateral movement and installed each one by one because the Powershell script file name, log file path, and file creation time were also different for each affected server.

C:\WINDOWS was mainly used for TinyPoS file creation and log file creation path.

Task data of TP-Case-1

In the PowerShell script, the TinyPoS is stored as a byte array data type, and the adversary added and used the WaitForSingleObject function for synchronization in the “shellcode_injection_expanded.powershell” of social-engineer-toolkit published on GitHub to load it into memory.

Powershell script of TP-Case-1

Most of the scripts contain only the 64bit version of TinyPoS, but some also include the 32bit version. Besides, there is a version that searches memory by targeting only the sqlservr.exe process, and all other than this searches memory based on the excluded process list.

[TP-CASE-2]

The second form of TinyPoS is to save as binary as a file. Since detection evasion techniques such as packing or obfuscation are not applied at all, this form is easy to detect by vaccines, and there is a high probability of being detected by engineers. The adversary created a file with the name “readme” disguised as an ordinary file, and it is presumed that TP-Case-2 was used only for the shortest time due to the risk of detection.

Unlike in TP-Case-1, the working path was changed to C:\ProgramData, and both 32bit and 64bit versions were included in a single file.

Press enter or click to view image in full size

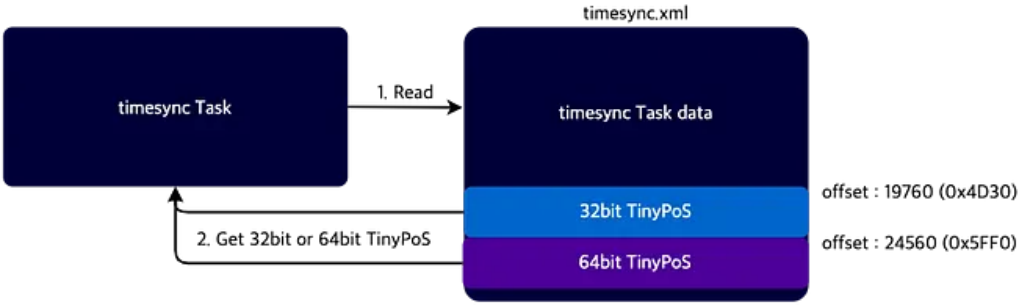
readme

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	55	89	E5	8B	45	08	EB	00	89	45	EC	8D	15	4F	10	40	U‰å‹E.ë.‰Eì..O.@
00000010	00	8D	05	08	10	40	00	83	E8	08	29	C2	8B	45	EC	01	.....@.ƒè.)Â‹Eì.
00000020	C2	31	DB	B8	0F	87	53	05	31	04	1A	81	3C	1A	90	90	Â1Û¸.‡S.1...<...
00000030	90	90	74	0D	83	FB	00	75	08	31	04	1A	40	EB	E9	EB	..t.ƒû.u.1..@ëéë
00000040	0E	83	C3	04	81	3C	1A	CF	CF	CF	CF	74	02	EB	D9	9F	.ƒÃ..<.ÏÏÏÏt.ëÙŸ
00000050	17	C3	95	9F	17	C3	95	9F	17	D0	E9	27	B6	81	61	84	.Ã•Ÿ.Ã•Ÿ.Ðé'¶.a„
00000060	D5	63	8E	5D	8B	D8	57	1B	0C	21	2D	B6	9F	53	05	0F	Õcž]‹ØW..!-¶ŸS..
00000070	B6	AC	34	CF	2B	6F	64	73	85	7F	25	CE	48	5E	04	C8	¶¬4Ï+ods….%ÎH^.È
00000080	65	A3	84	F0	DC	EF	4F	65	0C	09	15	84	95	26	DC	C8	e£„ðÜïOe...„•&ÜÈ
00000090	C2	A3	8B	41	89	BF	ED	48	96	53	05	86	C2	AF	C2	4A	Â£‹A‰¿íH–S.†Â¯ÂJ
000000A0	77	F9	F9	02	FB	BB	3D	1E	87	53	8C	4A	7F	94	40	FF	wùù.û»=.‡SŒJ.”@ÿ
000000B0	D3	99	AA	9E	6F	7A	14	0F	87	39	45	67	87	63	05	0F	Ó™ªžoz..‡9Eg‡c..
000000C0	EF	53	45	0E	87	39	05	F0	57	D8	58	F7	0E	0B	3D	84	ïSE.‡9.ðWØX÷..=„
000000D0	DA	AF	8C	57	B7	DA	C0	E7	8A	53	05	0F	EC	36	77	61	Ú¯ŒW·ÚÀçŠS..ì6wa
000000E0	E2	3F	36	3D	A9	37	69	63	87	AC	50	3F	0E	16	45	E7	â?6=©7ic‡¬P?..Eç
000000F0	8B	53	05	0F	F0	20	6A	6C	EC	60	37	21	E3	3F	69	0F	‹S..ð jlì`7!ã?i.
00000100	78	06	35	86	C2	1B	ED	03	87	53	05	4C	EB	3C	76	6A	x.5†Â.í.‡S.Lë<vj

[Figure 5] TP-Case-2 shellcode

[TP-CASE-3]

Press enter or click to view image in full size



[Figure 6] Overview of TP-Case-3 Execution Flow

The third form is to save the shellcode in an XML file disguised as a scheduler. When a task is registered in the task scheduler, a file in the same XML format as the task name is created in the path C:\WINDOWS\System32\Tasks. Settings related to work are saved in the file. The adversary registers a task that reads and executes timesync.xml under the name “timesync”.

The timesync.xml file was created by adding TinyPoS to the XML data related to the timesync task. The timesync.xml created by the adversary has the same data as the regular working file when opened with an editor. Since the added shellcode part is not visible in the form of a string.

Logfile that records the process of registering TP-Case-3 Scheduled Task

In TP-Case-3, unlike TP-Case-1, TinyPoS is loaded with a script partially modified from DKMC’s “exec-sc.ps1” published on Github.

Powershell script of TP-Case-3

Also, the offset is classified according to whether it is 64bit, and TinyPoS suitable for the target OS is read from timesync.xml. There have also been changes in the list of excluded processes. The list of excluded processes from other TP-Cases is mainly composed of general processes. However, in TP-Case-3, the adversary checked the collected data and its source process. After that, unnecessary processes inside the actual target server were included in the list of excluded processes.

timesync scheduled task

timesync.xml

[Overall Comparison of TP-Cases]

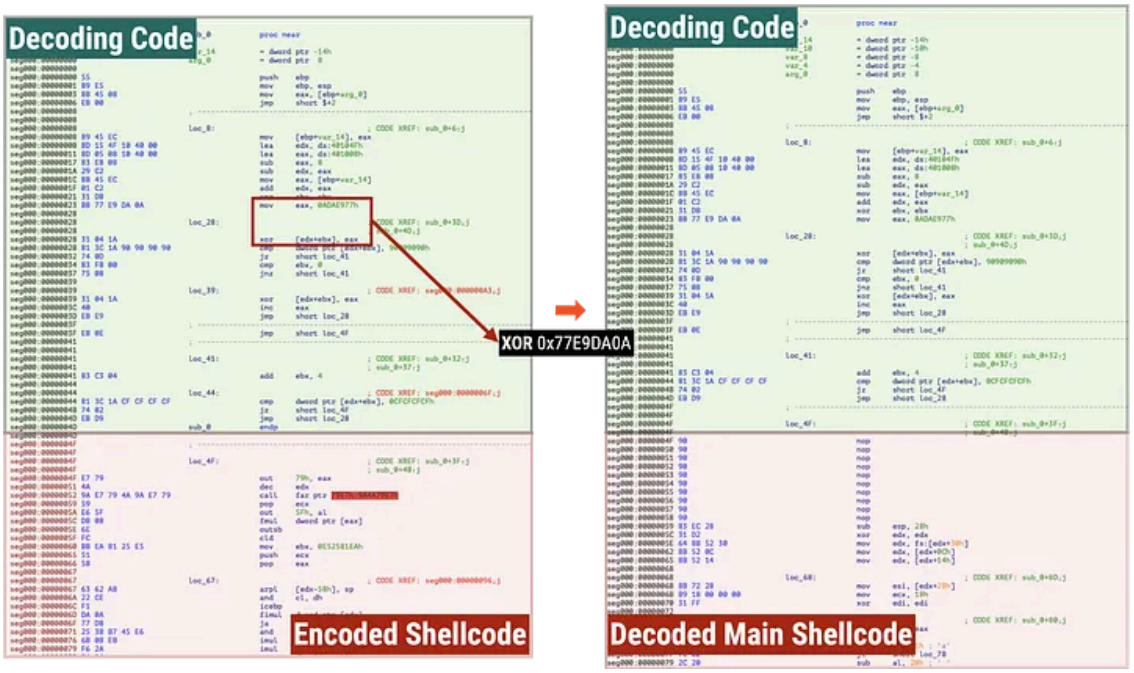
	TP-Case-1	TP-Case-2	TP-Case-3
File path	C:₩WINDOWS₩SqlSvc.ps1 C:₩WINDOWS₩SqlSrvTN.ps1 C:₩WINDOWS₩SqlClk.ps1 C:₩WINDOWS₩WavenSQL.ps1 C:₩WINDOWS₩S.ps1	C:₩ProgramData₩readme	C:₩ProgramData₩timesync.xml
File type	Powershell script	Binary file	XML file masquerading as task
Name of task	Same as filename		
When to use	2019/02~	2020/01~	2020/05~
Mutex name	kor9963 sqlsvc_t64 sqlcn_t32 sqlcn_t64	[ComputerName]	[ComputerName]
Decode key	-	0x0F875305	0x77E9DA0A
Check value	-	0xCFCFCFCF	0xCFCFCFCF
Logfile path	C:₩WINDOWS₩[ComputerName].kor C:₩WINDOWS₩log.kor C:₩WINDOWS₩kr.kor C:₩WINDOWS₩sqlite.log C:₩WINDOWS₩data.kor C:₩WINDOWS₩data.cn	C:₩ProgramData₩data_sync.db	C:₩ProgramData₩rj_log.dat
Logging format	[Track2 Data] *[Process]	[Track2 Data] *[Process] **xx[LocalIP]₩[ComputerName]	[Track2 Data] *[Process] **xx[LocalIP]₩[ComputerName]
Interval	6 hours	-	1 hour
List of excluded processes	alerts, alsvc., archiv, armsvc, boanet, busine, cisvc., clean., cmd.ex, conhos, csrss., dwm.ex, iastor, iexplo, inetin, java.e, lmigua, lms.ex, logmei, lsass., lsm.ex, ndagen, node.e, nssm.e, ppsgne, pxcont, python, ramain, safest, savadm, savser, sdcser, search, servic, shell., smss.e, snarec, sntpse, sophos, spools, sqlbro, sqlwri, sspser, svchos, swc_se, swi_se, syslog, tasken, taskho, timesr, uns.ex, update, winini, winlog, winvnc, wmiprv, xsauth, dllhos, excel., explor, mmc.ex, csrs.e, clamsc, regsvr, mobsyn, rundll, runonc, winwor, system, notepa, taskmg	alerts, alsvc., archiv, armsvc, boanet, busine, cisvc., clean., cmd.ex, conhos, csrss., dwm.ex, iastor, iexplo, inetin, java.e, lmigua, lms.ex, logmei, lsass., lsm.ex, ndagen, node.e, nssm.e, ppsgne, pxcont, python, ramain, safest, savadm, savser, sdcser, search, servic, shell., smss.e, snarec, sntpse, sophos, spools, sqlbro, sqlwri, sspser, svchos, swc_se, swi_se, syslog, tasken, taskho, timesr, uns.ex, update, winini, winlog, winvnc, wmiprv, xsauth, dllhos, excel., explor, mmc.ex, csrs.e, clamsc, regsvr, mobsyn, rundll, runonc, winwor, system, notepa, taskmg	alpbcs, armsvc, cntaos, conhos, csrss., dfs.co, dfssvc, dllhos, dwm.ex, explor, iastor, igfxcu, igfxem, iprose, iusb3m, jhi_se, lms.ex, lsass., lsm.ex, ntrtsc, pccntm, presen, ravbg6, rtkaud, search, servic, smss.e, spools, svchos, system, tasken, taskho, taskli, tmbmsr, tmccsf, tmlist, tmpfw., tt.exe, tvnser, wavess, wifiau, winini, winlog, wmiprv, wuaucl
Remarks	Sometiems TinyPos only monitored sqlsvr.exe (mssql)	-	-
Target expiration date	2018~2026 2019~2029	2020~2030	2020~2030

[Table 1] Comparison of Characteristics by TinyPos Installation Case

Detailed feature analysis

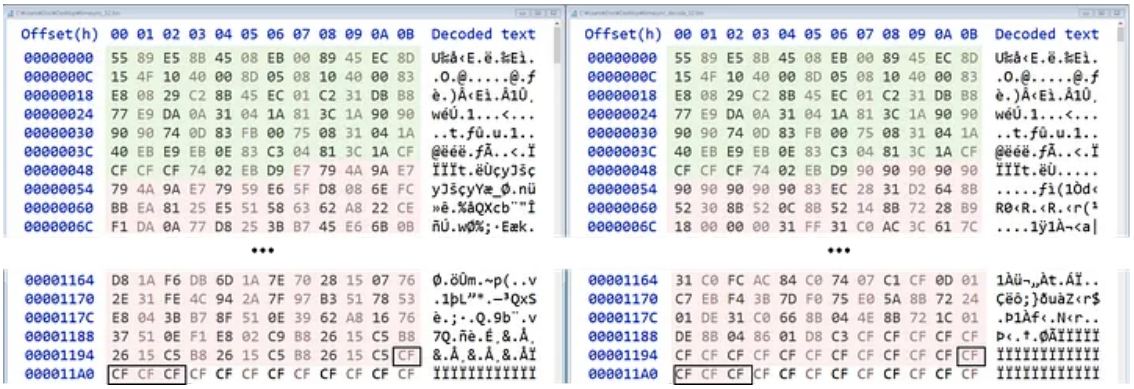
Only some codes were encoded in the early version of TinyPoS, but from TP-Case-2, the main code located at a specific offset is decoded by XOR and then executed. This process is repeated until a specific 4-byte value (0xCFCFCFCF) comes out. Afterward, a mutex is created to prevent duplicate execution.

Press enter or click to view image in full size



[Figure 7] Decoding process of TinyPos

Press enter or click to view image in full size



[Figure 7–1] Comparison of Encoded binary and Decoded binary

TinyPoS attempts to read all the running processes of memory except for specific processes listed by TinyPoS in order to collect the card data. When reading the memory, it only reads the page lists that matches a certain condition in table below. For TP-Case-2 and 3, PAGE_READWRITE and MEM_PRIVATE are not checked.

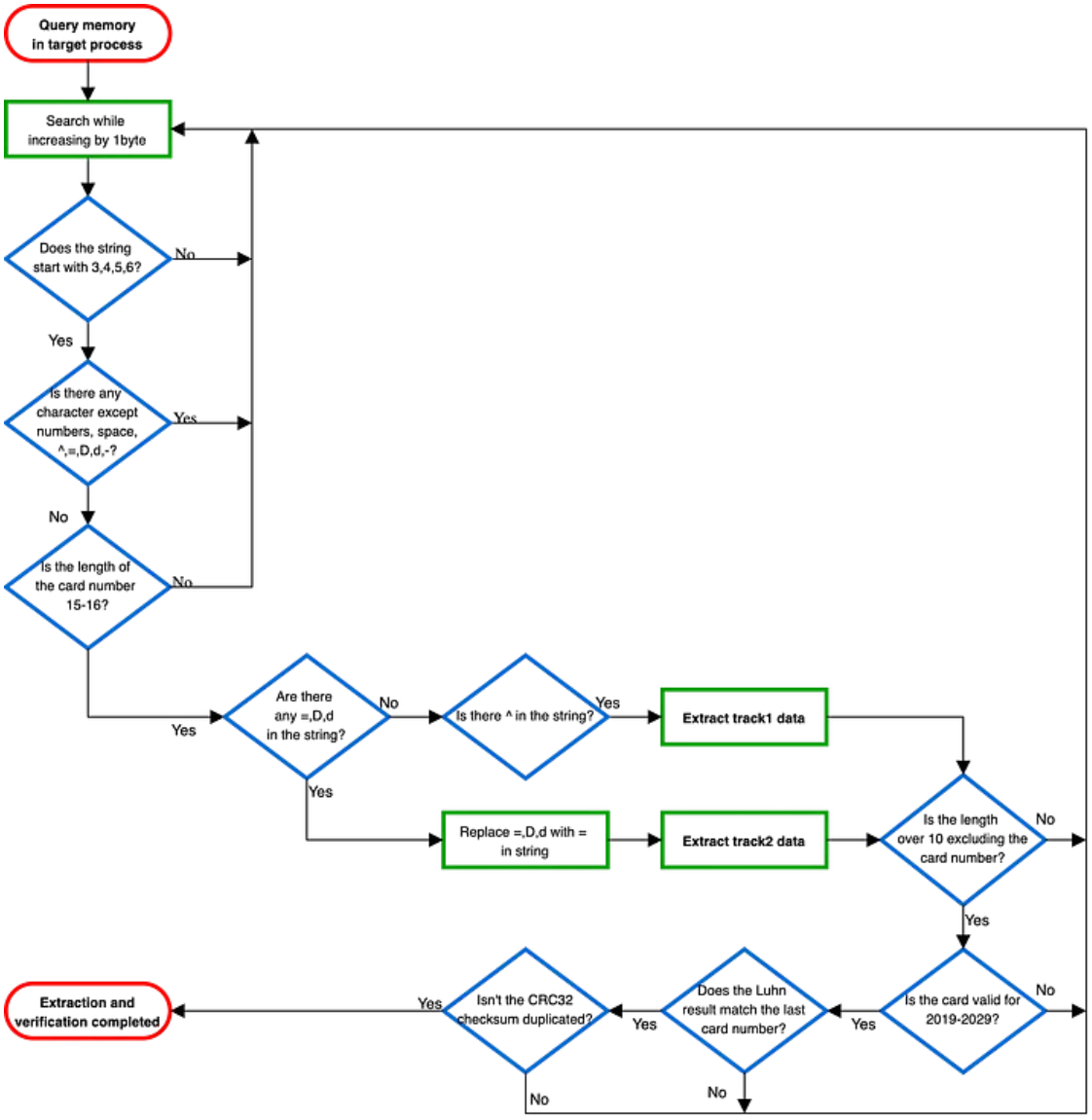
Press enter or click to view image in full size

Field	Condition	Case	Description
State	== MEM_COMMIT (0x1000)	TP-Case-1, 2, 3	The real physical memory of the page Check the mapping status (check whether it is in use)
Protect	!= PAGE_NOACCESS (0x1)	TP-Case-1, 2, 3	Check if the page is inaccessible
Protect	!= PAGE_GUARD (0x100)	TP-Case-1, 2, 3	Check if the page is protected
Protect	== PAGE_ READWRITE (0x4)	TP-Case-1	Check whether the page can be read and write
Type	== MEM_PRIVATE (0x2)	TP-Case-1	Check whether the page is shared with other processes

[Table 2] Memory property to be scraped

The flow chart of the track 1 and 2 data extraction and verification process in memory is described in below. When searching Track 1, for a wider search, it searches for 120 bytes, which is 40 bytes larger than the maximum length defined in the standard, and Track 2, it searches for 40 bytes, which is the same as the standard maximum length.

Press enter or click to view image in full size



[Figure 8] Overall track data extraction and verification process

After going through the above process, refined data containing actual information is extracted, excluding the starting characters such as '%' or ';' and the LRC value. After that, set the buffer in the format of '[Track Data] * [Process]**xx[LocalIP]\[ComputerName]' by combining the local IP of the running server and the computer name, and then XOR-encode the data in units of 8 bytes using the value 0x6d2a1f3cb26e0c9f as a key. An 8-byte identification value of 0x20202020DD0ADD0A is added to the finally encoded data, then saved in a log file. After that, the next memory is searched and the process of extracting and verifying data in Track 1 and 2 is repeated.

As a result of comparing various versions, an adversary's mistake was found in the initial code.

In TP-Case-1, the search length of Track 1 was 60 bytes, and the search length of Track 2 was 120 bytes. It was reversed in size and was fixed in a later version.

We've noticed parser error of card data as it counts the space in between the strings of 15 to 16 numbers. For instance, if there is '[8 numbers][multiple spaces][7 numbers]=', this pattern is recognized as a card number or card data that successfully runs through the validation process and stored in the log file of TinyPoS.

Possibility of Exfiltration

An attempt to steal card data targeting database-related processes such as MICROS Database Service (resdbs.exe), SQL Server (sqlsvr), and MySQL (mysqld) was already used by TinyPoS which implies that this method of exfiltration has been adopted at some point in the past.

Even if a specific data is deleted through a database query, depending on the type of DB, the data deleted by the query may still remain in the DB file, such as “mdf” for MS-SQL. The adversary already recognizes these characteristics and attempts to exfiltrate. Thus, in the case of TA505, we firmly believe that this notorious threat actor behind the clop ransomware recognized the above circumstances.

Therefore, we advise in the case of DBA (Database Administrator) dealing with sensitive data, it is necessary to understand and recognize these databases’ forensic characteristics when operating the system. All event logs are deleted due to event log deletion feature by clop ransomware making hard for scrutinizing forensic evidence, but we’ve noticed that the adversary set their own base server inside and installed TinyPoS on each server using PsExec initially and TinyMet afterward, and finally collected sensitive data.

Possibility of Misuse of the Stolen Card Information

Track 1 and 2 data standards follow the international standard ISO/IEC 7813. According to this standard, items such as PVKI, PVV, and CVC are included in the Discretionary Data part. In practice, it is structured in the form of a combination of the cardholder verification code and a one-way hash value generated by combining information such as card number and expiration date.

Also, the CVV (Card Verification Value) stored in Track 2 is a number for checking the integrity of Track data, called CVV1, and has a different value from CVV2 marked on the actual card used for online transactions. Therefore, in most online shopping malls that require CVV2, making a payment with only Track 2 data is impossible. Some shopping malls do not request CVV2 information as in quick payment, but there must be additional information given in alternation with not providing CVV2.

Get S2W’s stories in your inbox

Join Medium for free to get updates from this writer.

Remember me for faster sign in

However, after going through some processes with Track 2, it is possible to use the duplicated card offline. Special hardware equipment and programs are required to duplicate a card, which can be requested to a third party through Deep Web/Dark Web. Since there are flooding number of carding forums already exist and are active, this is not bothered at all. Therefore, the card information is one of the data actively traded. Card data stolen by an adversary can be exploited directly, but it can be sold on dark web forums or carding forums to reduce the risk of using duplicate cards offline. If data is sold in this way, since it is leaked to a person who is willing to abuse it, payment by a third party may occur afterward, resulting in actual damage due to the card leakage accident.

Correlation Analysis

Clop

In the early days of Clop ransomware, CobaltStrike and FlawedAmmyy were used. FlawedAmmyy is a remote control tool developed by exploiting a specific remote management solution's source code. It has been used before TA505 that has been utilized as a Clop ransomware attack; nevertheless, both FlawedAmmyy and Clop ransomware has been packed by the identical packing, also a few features of malicious code was signed by an identical certificates.

Afterwards, the attack was attempted using Amadey Bot, SDBbot, Get2, FlawedGrace, etc. Clop ransomware targets the AD environment and is distributed in bulk after taking over administrator privileges.

Azorult

Azorult, which has been packed with the same custom packer as Clop ransomware, also has a function to steal credit card information. However, Azorult focuses on collecting digital footprint data such as infected device information, browser information, and coin wallet information rather than credit card information itself. In addition, since credit card information is also collected from the data stored in the Chrome browser through the query seen in the below, it performs slightly different role than TinyPoS intensively steals Track 1 and 2 data.

```
SELECT name_on_card, expiration_month, expiration_year, card_number_encrypted value FROM credit_card
```

AbaddonPoS & PinkKite

AbaddonPoS is a PoS malware coined its term by Proofpoint in November 2015. On January 2016, Foregenix reports a malware called TinyPoS, and the known difference in functionality between the two malware was not significant. Only difference was the way of implementation; AbaddonPoS has the shellcode encoded inside, while TinyPoS did not encoded. In addition, TinyPoS controls the thread branch with the values 0xC0C0C0C and 0xF0F0F0F. This has been also mentioned by Trendmicro through the report released on May 2016.

A similar PoS malware, called PinkKite, was issued in 2018, but in April 2019, Forcepoint's report stated that the difference between TinyPoS and PinkKite seem very similar except whether the leaked information is flowed to the network or stored as a file. Further analysis was not doable due to lack of PinkKite hash. In other words, AbaddonPoS and PinkKite can be seen as the same cluster of malware as TinyPoS. It was also referred to as "TinyPoS or PinkKite" in a report by CarbonBlack in May 2020.

Initially some differences prevailed, however making the distinction between TinyPoS and AbaddonPoS lacked a point after Proofpoint stating TinyPoS as a type of AbaddonPoS.

After 2018, this difference has disappeared and is used in the same manner as TP-Case-1, 2, and 3 as mentioned above implicating that TA505 is likely to have purchased and used the TinyPoS from a third party. According to CrowdStrike's report, the attack group that produced TinyLoader and TinyPoS was named Tiny Spider and classified them as a separate group.

	AbaddonPoS – Proofpoint	TinyPoS - TrendMicro
MD5	a168fef5d5a3851383946814f15d96a7 7c7801f77fe6866367284914fd3c3b02	deb132c28f43fd86508f5ef363a28a73 67ec2c79e0ab81d2399650f30198d1e8
Decoding method	XOR Key Bruteforce	-
Encoding Key	0x2211AAFF	0x4C5D6E7F 0xCAEF3D8A
FS	^, =, D (Added later)	^, =, D, d (Added later)
Leakage path	Network	Network
Target card number length	13~19 13~16	13~16
Target expiration date	2015~2027 2015~2026	2015~2026 2017~2026
De-duplication method	String comparing	String comparing CRC32
Logging format	[Track] ***[ProcessName]	[Track] *[ProcessName]
List of excluded processes	svchos System smss.e explor csrss. winlog lsass. spools alg.ex winini steam. skype. dwm.ex Search taskho rundll cmd.ex lsm.ex dllhos regsvr regedi conhos	.cmd conhost. dllhost. EXCEL.ex explorer lsass.ex mmc. dwm. csrs igfx winlogon clamscan regsvr32 mobsync. rundll32 runonce. services spoolsv. svchost. taskhost taskmgr. WINWORD.

[Table 3] Comparison of AbaddonPoS and TinyPos analyzed by Trendmicro

TinyLoader

TinyLoader is a downloader malware released by Proofpoint along with TinyPoS (aka. AbaddonPoS). TinyLoader was used as a downloader for TinyPoS for approximately 4 years until the end of the year 2019. TinyLoader is composed of the same anti-debugging and obfuscation code as TinyPoS, and according to TrendMicro, each module that performs screen capture and process information collection can be additionally downloaded from the C&C server. The additional module download method is performed through the payload in charge of HTTP communication, and TinyPoS is also downloaded in the identical method.

According to Talos' report released in November 2019, DoppelPaymer, TinyPoS, SVCHOST SAMPLE, etc., were distributed from the same server. As a result of our analysis, SVCHOST SAMPLE was identical to TinyLoader. That is, TinyLoader, TinyPoS, and DoppelPaymer were distributed together from one server. At that time, the distributed TinyPoS was TP-Case-1. Through this, it was confirmed that the Powershell-type TinyPoS execution method was used in other incidents after at least April 2019.

Since TinyLoader is also distributed from the same server, it is assumed that TinyPoS of TP-Case-1 is also installed through TinyLoader. Furthermore, PsExec was discovered from the same server. For that reason, the attack flow at that time was as follows.

> **PsExec → [TinyLoader] → TinyPoS → DoppelPaymer**

However, according to the result of our recent incident response, the attack flow has been changed slightly as follows.

> **PsExec → [TinyMet or CobaltStrike] → TinyPoS → Clop ransomware**

In addition, it was confirmed that Vawtrak derived from Gozi downloaded TinyLoader, and it has evolved into malware called IcedID.

According to CrowdStrike, the IcedID malware is currently run by the Lunar Spider, and it has been confirmed that TinyLoader was distributed from the IcedID in 2019 following the Vawtrak malware in 2015.

DoppelPaymer

According to Talos's report mentioned above, DoppelPaymer ransomware was also distributed from the same server. DoppelPaymer is very similar in source code to BitPaymer ransomware.

ESET reports that BitPaymer and Dridex malware are related in several ways. As a result, it was revealed that the same developer produced the two different malware. Also, the analysis results of CrowdStrike show that Dridex malware and BitPaymer ransomware were found in identical incidents which strengthens the fact that it must be the same developing group.

8 months later, an analysis report released by CrowdStrike revealed that DoppelPaymer was derived from BitPaymer. Some Evil Corp personnel that developed Dridex were separated into a threat group called Doppel Spider and attempted an attack using DoppelPaymer. It is also known that the group developed DoppelDridex, known as Dridex 2.0 version. The overall information can be seen in CERT-FR's CTI report.

Currently, Evil corp is mainly used to refer to the Indrik Spider, and it attempts to attack using WastedLocker instead of the BitPaymer. No assurance can be stated that Doppel Spider has purchased these tools from Tiny Spider that operates TinyLoader and TinyPoS, however, we can assume close relation between the two organizations by the above-mentioned facts.

ProLock

After TP-Case-1 was mentioned in the November 2019 Talos report, TinyPoS reappeared in the report released by CarbonBlack in May 2020. Although the PowerShell script is slightly different, it is assumed to be of the TP-Case-3 type. Malware mentioned in this report also appeared in VISA analysis report on September 2020. Through this, we assume that the change was made from TP-Case-1 to TP-Case-3 over the course of six months.

Norfolkinfosec confirmed the above information and conducted further analysis, revealing that the decoding code pattern of TinyPoS is the same as that of ProLock ransomware except the check value of TinyPoS is '0xCFCFCFCF', while ProLock is '0xC4C4C4C4'. The IoC released by Norfolkinfosec contains ProLock malware with the same file name as the "readme" of TP-Case-2 described in this report.

Group-IB released its analysis report for ProLock in May 2020. The PowerShell script that executes ProLock is almost the same as the one used when executing TP-Case-3. And, the batch scripts that register the scheduler task in TP-Case-1 and 3 are also mentioned in the Talos report.

Finally, we have confirmed the similar features between two malware, ProLock and TinyPos.

1. Primary work path (C:\ProgramData)
2. Anti-debugging technique
3. Decoding code
4. Hard-coding pattern of search target process list
5. Process name comparison code
6. Initially Distributed in PE form and then as a binary blob
7. The use of '*' for buffer initialization (ProLock) and delimiter (TinyPoS)

In conclusion, it is highly likely that these results are derived from the same developer. We can assume that Tiny Spider has also started the ransomware business with ProLock but the client seems to moved to Egregor ransomware due to issues associated with ProLock.

Overall Connection

Conclusion

As the spread of IC (integrated circuit) cards increases, the damage to companies or financial sectors caused by PoS malware is gradually decreasing. However, card data that cannot be identified where it was stolen is being traded on the DDW (Deepweb and Darkweb). Even if a firm securely manages the card data and the related device, if the backed-up data in the past is not properly managed, an adversary can target this gap and steal important information. Usually, card data theft is secretly and continuously performed so that victims cannot notice it immediately, so it is not easy to determine whether there is an infection.

TinyPoS is frequently found in many other card data theft incidents as well as ransomware incidents related to DoppelPaymer and Clop. In this attack, the adversary continuously used various versions of TinyPoS, updating the code in a frequent manner, such as changing the list of processes to be collected and amending the minor mistakes in a prompt manner. It means that the developers of TinyPoS have sold the source code or are working closely with buyers.

Countless cybercrime threat actors are already increasing the cybercrime scale by cooperating by selling each other's resources. As ransomware attack organizations increase and the cyber black market grows, cooperation among these criminal organizations will continue.

Cybercriminals encrypt crucial files in the enterprise, by extorting money as hostages for stolen confidential information or customer data. These ransomware gangs will come up with various ways to acquire cash besides the existing attack methods using only ransomware. In the early days of ransomware, encrypted files were used as bait. Since then, the entire MBR was encrypted, the negotiation amount increased over time, targeting companies with relatively sufficient funds and attacking high-level executives. Various strategies are emerging each time. Most recently, there have been threats to attempt a DDoS attack if the negotiations are not accepted.

The hijacking of card data is one of these strategies, and if it is successful, the adversary can request a higher amount of money from the victim. Even if negotiations fail, the stolen card data can benefit. In addition to TA505

in this report, FIN6 also has malware such as Ryuk, LockerGoga ransomware, and GratefulPoS. We confirmed that Revil (Sodinokibi) operator is also interested in PoS software.

Ransomware gangs that are good at infiltrating and stealing data continue to stay within the enterprise and seek valuable assets. They will use a variety of strategies to monetize these data. These attempts will increase in the future, and ultimately, adversaries will continue to think about ways to make more money.
