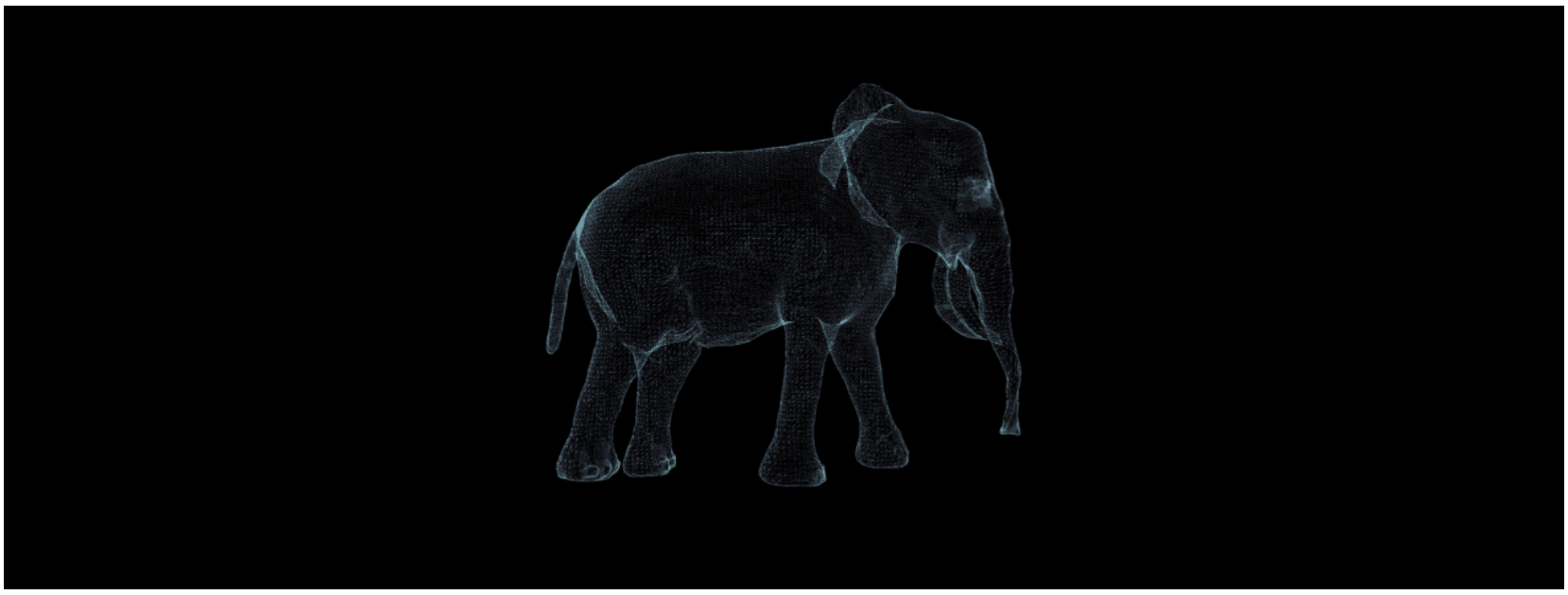
# Elephant Framework Delivered in Phishing Attacks Against Ukrainian Organizations

Written by Joakim Kennedy and Nicole Fishbein - 4 April 2022



A recently developed malware framework called Elephant is being delivered in targeted spear phishing campaigns using spoofed Ukrainian governmental email addresses. The four malware components delivered are used for stealing credentials, documents, and to provide remote access to the infected machine.

Two of these components were first reported on by the Computer Emergency Response Team for Ukraine (CERT-UA) in March 2022. They named the two components GraphSteel and GrimPlant. When investigating these events, we have identified that Elephant has also been delivered via phishing emails from spoofed Ukrainian email addresses. Elephant is a malware framework written in Go. The activity has been attributed to UAC-0056 (TA471, SaintBear, UNC2589) by CERT-UA.

## Background

On March 12, the CERT-UA published an alert on a threat about phishing emails sent on behalf of state bodies of Ukraine that urged the recievers to update the system by using a link provided in the email. Once the user opens the link two files are downloaded, one is Cobalt Strike Beacon the other is a dropper that will download and execute two additional files. These additional files are base64 encoded, the files saved as: “microsoft-cortana.exe” (classified as GraphSteel) and “oracle-java.exe” (GrimPlant backdoor). The attack used Discord as a hosting server for the additional payload that was downloaded.

On March 15, SentinelOne found two more samples of the GraphSteel and GrimPlant malware families. These samples written in Go were dropped by an executable that was disguised as a translation application. The new samples are similar to those published by the CERT-UA, both in the naming and the functionality.

On March 28, CERT-UA published another alert about phishing emails with an attached “xls” file. The subject of the email and the file name are both “Wage arrear”. The attached file had macros that, once executed, created a file called Base-Update.exe — a malware that downloads and executes two other files classified as GraphSteel and GrimPlant by the CERT.

## Phishing Email

- The subject of the email: Заборгованість по зарплаті (arrears in wages).
- The email was sent from zam@mdfi.gov.ua to ilenko@gng.com.ua.
- The sender domain “mdfi.gov.ua” belongs to the Mykolayiv Regional Phytosanitary Laboratory.
- The receiver is the Head of Department of Technical Supply in OKKO Group, one of the largest filling stations in Ukraine, this person is in charge of the gas stations chain.
- Based on the headers of the email: the email was sent from 87.249.139.161 (hosting web server located in Turkey).

```
Return-Path: <zam@mdfi.gov.ua> Received: from hosting30.ukrnames.com (hosting30.ukrnames.com
[217.182.197.11]) by mx-fm0.gng.com.ua with ESMTP id 22RJjl16017368-22RJjl18017368 (version=TLSv1.2
```

cipher=ECDHE-RSA-AES256-GCM-SHA384 bits=256 verify=NO) for <ilenko@gng.com.ua>; Sun, 27 Mar 2022 22:45:47 +0300 Received: from [87.249.139.161] (port=15731 helo=WIN3ISR1T95E6Qwwwtendawificom) by hosting30.ukrnames.com with esmtpsa (TLS1.2) tls TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (Exim 4.95) (envelope-from <zam@mdfi.gov.ua>) id 1nYYot-00AQaf-Gj for ilenko@gng.com.ua; Sun, 27 Mar 2022 22:45:47 +0300 MIME-Version: 1.0 From: "zam@mdfi.gov.ua" <zam@mdfi.gov.ua> Reply-To: zam@mdfi.gov.ua To: ilenko@gng.com.ua ... X-Mailer: Smart_Send_4_4_2 Date: Sun, 27 Mar 2022 12:44:30 -0700 Message-ID: <1278047665520817181238 41@WIN-3ISR1T95E6Q> X-AntiAbuse: This header was added to track abuse, please include it with any abuse report X-AntiAbuse: Primary Hostname - hosting30.ukrnames.com X-AntiAbuse: Original Domain - gng.com.ua X-AntiAbuse: Originator/Caller UID/GID - [47 12] / [47 12] X-AntiAbuse: Sender Address Domain - mdfi.gov.ua X-Get-Message-Sender-Via: hosting30.ukrnames.com: authenticated_id: zam@mdfi.gov.ua X-Authenticated-Sender: hosting30.ukrnames.com: zam@mdfi.gov.ua X-Source: X-Source-Args: X-Source-Dir: X-FE-Attachment-Name: =?UTF-8?B?x+Dh7vDj7uLg7bPx8vwg7+4g5+Dw7+vg8rMueGxz?= X-FEAS-SBL: 87.249.139.161 score 1 X-FE-Policy-ID: 2:1:4:gng.com.ua X-FE-Orig-Env-Rcpt: ilenko@gng.com.ua X-FE-Orig-Env-From: zam@mdfi.gov.ua X-FEAS-Client-IP: 217.182.197.11

While reviewing the email's transport path, we noticed that the domain "mdfi.gov.ua" did not have a configured SPF record to prevent email spoofing. An SPF record is used to restrict which IP addresses are allowed to send emails for a specific domain. If this record is not set, any IP address is technically allowed to send emails using that domain name. Some email providers do warn when they receive an email from an address that doesn't have an SPF record. The screenshot below shows the warning message displayed in GMail when reading such email.


Our theory is that the threat actor exploited this vulnerability to send spoofed phishing emails to their targets. We reported the issue to CERT-UA.

We followed the sender IP address and found three other emails: two emails submitted on March 29 from Ukraine targeting ICTV, a Ukrainian TV channel and the third email was submitted in February from Romania. The emails that target UA have the same subject and use the same attached xls file that delivers the first malicious payload.

## The Elephant Framework

The malware that is dropped by the phishing lure is the dropper component of what we call the "Elephant Framework." The framework consists of four components that work in unison. The code snippet below shows a reconstruction of the source code tree, bold indicating folders, showing how the different components have been organized. The location of the implant's entrypoint is unknown and has been guessed to be in the root folder. As there are also server components to the framework, we hypothesize that there are more folders for the two servers used by the framework.

```
elephant
   client
      cmd
         implant
            main.go
      client.go
      strings.go
   downloader
      cmd
         downloader
            main.go
      downloader.go
      strings.go
   dropper
      cmd
         dropper
            main.go
      dropper.go
      strings.go
   internal
      implant
         auth_client.go
         auth_interceptor.go
         dialoptions.go
         implant.go
         settings
         settings.go
   proto
      implant
         implant.pb.go
         implant_grpc.pb.go
   (implant.go)
```

```
elephant
   client
      cmd
         implant
            main.go
      client.go
      strings.go
   downloader
      cmd
         downloader
            main.go
      downloader.go
      strings.go
   dropper
      cmd
         dropper
            main.go
      dropper.go
      strings.go
internal
   implant
      auth_client.go
      auth_interceptor.go
      dialoptions.go
      implant.go
      settings
      settings.go
proto
   implant
      implant.pb.go
      implant_grpc.pb.go
(implant.go)
```

## Dropper Component

While called the dropper in the framework, this component does not have an embedded payload. Instead it is technically a downloader that fetches the next stage called the "downloader." The next stage is downloaded from the URL "hxxp://194.31.98.124:443/i" and saved to the user's home directory (%HOME%/.java-sdk/java-sdk.exe). The next stage is executed with the command line flag "-a 0CyCcrhI/6B5wKE8XLOd+w==", base64 and AES encrypted information about the C2 server.

## Downloader Component

The "downloader" acts as an orchestrator for the other components. In addition to downloading the "client" and the "implant" components, it also sets up persistence and can perform updates. Like all the other components, before any malicious activity is taken it performs some evasion techniques. The difference between this component and the others is that this one is using code from the ColdFire project on GitHub. The screenshot below shows the malware using the "Wait" function to sleep for 10 seconds before allocating 200 mb of garbage data.

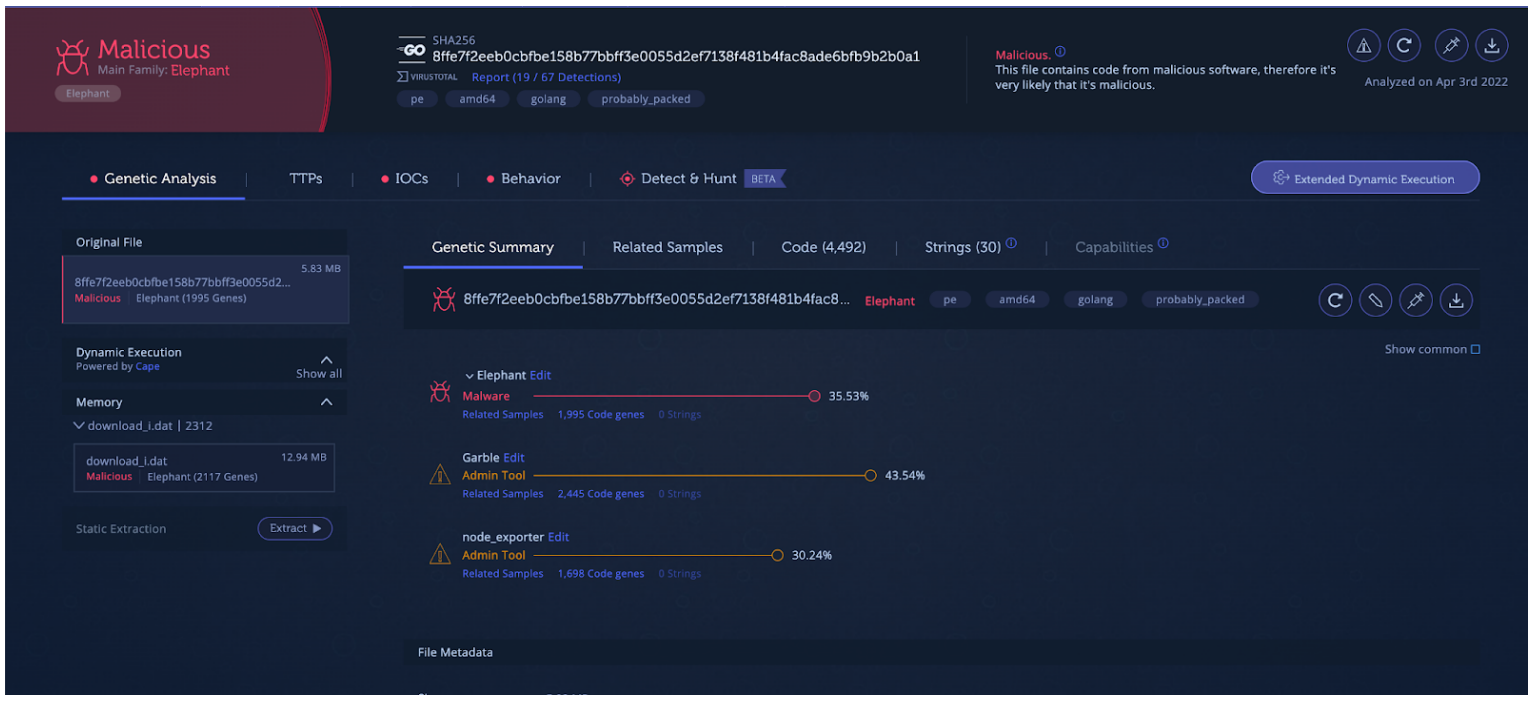
```
94: fcn.elephant_downloader.ev ();
        ; var int64_t var_18h @ rsp+0x18
      ┌─> 0x0069ee20      493b6610        cmp rsp, qword [r14 + 0x10]
      ┌─< 0x0069ee24      7651            jbe 0x69ee77
      │   0x0069ee26      4883ec20        sub rsp, 0x20
      │   0x0069ee2a      48896c2418      mov qword [var_18h], rbp
      │   0x0069ee2f      488d6c2418      lea rbp, [var_18h]
      │   0x0069ee34      488d05e32b09.   lea rax, [0x00731a1e]       ; "10s12516020020420630440040404
      │   0x0069ee3b      bb03000000      mov ebx, 3
      │   0x0069ee40      e85bd3ffff      call fcn.github.com_redcode_labs_ColdFire.Wait ;[1]
      │   0x0069ee45      488d05b48a02.   lea rax, [0x006c7900]
      │   0x0069ee4c      bb0000800c      mov ebx, 0xc800000
      │   0x0069ee51      4889d9          mov rcx, rbx
      │   0x0069ee54      e867a7daff      call fcn.runtime.makeslice  ;[2]
      │   0x0069ee59      31c9            xor ecx, ecx
    ┌─< 0x0069ee5b      eb07            jmp 0x69ee64
    ┌─> 0x0069ee5d      c60408ff        mov byte [rax + rcx], 0xff    ; [0xff:1]=255 ; 255
    ╎╎  0x0069ee61      48ffc1          inc rcx
    ╎╎  ; CODE XREF from fcn.elephant_downloader.ev @ 0x69ee5b
    └─> 0x0069ee64      4881f9ffff7f.   cmp rcx, 0xc7fffff
    └─< 0x0069ee6b      7ef0            jle 0x69ee5d
      │ 0x0069ee6d      488b6c2418      mov rbp, qword [var_18h]
      │ 0x0069ee72      4883c420        add rsp, 0x20
      │ 0x0069ee76      c3              ret
      └─> 0x0069ee77      e86417dcff      call fcn.runtime.morestack_noctxt ;[3]
      └─< 0x0069ee7c      eba2            jmp fcn.elephant_downloader.ev
```


```
94: fcn.elephant_downloader.ev ();
        ; var int64_t var_18h @ rsp+0x18
      ┌─> 0x0069ee20      493b6610        cmp rsp, qword [r14 + 0x10]
      ┌─< 0x0069ee24      7651            jbe 0x69ee77
      │   0x0069ee26      4883ec20        sub rsp, 0x20
      │   0x0069ee2a      48896c2418      mov qword [var_18h], rbp
      │   0x0069ee2f      488d6c2418      lea rbp, [var_18h]
      │   0x0069ee34      488d05e32b09.   lea rax, [0x00731a1e]       ; "10s12516020020420630440040404
      │   0x0069ee3b      bb03000000      mov ebx, 3
      │   0x0069ee40      e85bd3ffff      call fcn.github.com_redcode_labs_ColdFire.Wait ;[1]
      │   0x0069ee45      488d05b48a02.   lea rax, [0x006c7900]
      │   0x0069ee4c      bb0000800c      mov ebx, 0xc800000
      │   0x0069ee51      4889d9          mov rcx, rbx
      │   0x0069ee54      e867a7daff      call fcn.runtime.makeslice  ;[2]
      │   0x0069ee59      31c9            xor ecx, ecx
    ┌─< 0x0069ee5b      eb07            jmp 0x69ee64
    ┌─> 0x0069ee5d      c60408ff        mov byte [rax + rcx], 0xff    ; [0xff:1]=255 ; 255
    ╎╎  0x0069ee61      48ffc1          inc rcx
    ╎╎  ; CODE XREF from fcn.elephant_downloader.ev @ 0x69ee5b
    └─> 0x0069ee64      4881f9ffff7f.   cmp rcx, 0xc7fffff
    └─< 0x0069ee6b      7ef0            jle 0x69ee5d
      │ 0x0069ee6d      488b6c2418      mov rbp, qword [var_18h]
      │ 0x0069ee72      4883c420        add rsp, 0x20
      │ 0x0069ee76      c3              ret
      └─> 0x0069ee77      e86417dcff      call fcn.runtime.morestack_noctxt ;[3]
      └─< 0x0069ee7c      eba2            jmp fcn.elephant_downloader.ev
```


Persistence is established by adding a new key entry with the name “Java-3DK” to the registry key “Software\Microsoft\Windows\CurrentVersion\Run”. After this, the malware checks if a new binary exists by comparing its MD5 hash with a hash from the server. If no update is needed, it downloads the other two components.

The downloader has some 3rd party libraries, whose metadata are listed in the binary, that are not used. All the 3rd party libraries are listed in the code snippet below. In the list for example “port-scanner”, “gopacket”, and “gateway” packages are not being used. These are all libraries to facilitate lateral movement. It is not clear if these are left-overs from an older version of the malware or hints of future functionality.

```
github.com/anvie/port-scanner v0.0.0-20180225151059-8159197d3770 github.com/dustin/go-humanize v1.0.0
github.com/fatih/color v1.10.0 github.com/google/gopacket v1.1.19 github.com/google/gopacket/layers v1.1.19
github.com/google/gopacket/pcap v1.1.19 github.com/jackpal/gateway v1.0.7 github.com/jcmturner/aescts
v2.0.0+incompatible github.com/mattn/go-colorable v0.1.8 github.com/mattn/go-isatty v0.0.12 github.com/
minio/minio/pkg/disk v0.0.0-20210213070509-a94a9c37faf5 github.com/mitchellh/go-ps v1.0.0 github.com/
redcode-labs/ColdFire v0.0.0-20210118141151-d4d62410b029 github.com/savaki/jq
v0.0.0-20161209013833-0e6baecebbf8 github.com/savaki/jq/scanner v0.0.0-20161209013833-0e6baecebbf8
golang.org/x/sys/windows v0.0.0-20210119212857-b64e53b001e4 golang.org/x/sys/windows/registry
v0.0.0-20210119212857-b64e53b001e4
```




Malicious
Main Family: Elephant
Elephant
SHA256
8ffe7f2eeb0cbfbe158b77bbff3e0055d2ef7138f481b4fac8ade6bfb9b2b0a1
VIRUSTOTAL Report (19 / 67 Detections)
pe
amd64
golang
probably_packed
Malicious.
This file contains code from malicious software, therefore it's very likely that it's malicious.
Analyzed on Apr 3rd 2022
Genetic Analysis
TTPs
IOCs
Behavior
Detect & Hunt
BETA
Extended Dynamic Execution
Original File
5.83 MB
8ffe7f2eeb0cbfbe158b77bbff3e0055d2...
Malicious
Elephant (1995 Genes)
Dynamic Execution
Powered by Cape
Show all
Memory
download_i.dat | 2312
download_i.dat
12.94 MB
Malicious
Elephant (2117 Genes)
Static Extraction
Extract
Genetic Summary
Related Samples
Code (4,492)
Strings (30)
Capabilities
8ffe7f2eeb0cbfbe158b77bbff3e0055d2ef7138f481b4fac8...
Elephant
pe
amd64
golang
probably_packed
Show common
Elephant Edit
Malware
35.53%
Related Samples
1,995 Code genes
0 Strings
Garble Edit
Admin Tool
43.54%
Related Samples
2,445 Code genes
0 Strings
node_exporter Edit
Admin Tool
30.24%
Related Samples
1,698 Code genes
0 Strings
File Metadata

[Analysis](#) of the Elephant downloader

# Implant Component (GrimPlant)

GrimPlant is a backdoor that allows the operator to execute arbitrary PowerShell scripts on the infected machine. The backdoor has a relatively small set of functionality, for example it doesn't have any persistence functionality on its own. When the malware first is executed, it allocates 200 mb and sleeps for 10 seconds, the function shown in the screenshot below. This is an anti-emulation technique that has been found in other malware written in Go.


The Command and Control (C2) address is not included in the binary. Instead it is passed in to the malware via the command line flag “-addr”. The address is not provided as a plain string. Instead it has been encrypted with AES in Cipher-Block Chaining (CBC) mode. The malware decrypts the string with the embedded key (f1d21960d8eb2fddf2538d29a5fd50b5f64a3f9bf06f2a3c4c950438c9a7f78e) and a null IV. The port used by the C2 server is hardcoded to port 80.

GrimPlant communicates with the C2 server over gRPC. The communication is encrypted with TLS. The malware has an embedded root certificate that it uses to verify that it talks to a trusted server. The code snippet shows parts of the root certificate information. The certificate used by the C2 server has been signed by this root certificate which allows the threat actor to rotate the certificate without redeploying a new malware.

```
Version: 3 (0x2) Serial Number: 6d:56:93:aa:f3:9d:b1:f7:15:4e:39:64:77:9c:7e:d0:d4:cf:f6:3e Signature Algorithm: sha256WithRSAEncryption Issuer: C = FR, ST = Occitanie, L = Toulouse, O = O, OU = E, CN = *.a.com, emailAddress = a@mail.com Validity Not Before: Mar 20 15:21:06 2022 GMT Not After : Mar 20 15:21:06 2023 GMT Subject: C = FR, ST = Occitanie, L = Toulouse, O = O, OU = E, CN = *.a.com, emailAddress = a@mail.com SHA1 Fingerprint=DC:A9:5B:DC:F0:53:55:73:7A:A6:79:85:43:F6:3E:7C:23:07:36:33
```

There are only a handful of gRPC “methods” supported by the malware. A reconstructed protobuf specification is shown in the code snippet below. To identify which instance of the malware is sending the request to the C2 server, the malware uses its machine ID as an unique identifier in the messages. When the malware first connects to the C2 server, it authenticates itself with the password “sdrunlygvhwbcaeiuklgunvre”.

After a successful authentication, it sends a heartbeat message every 10 seconds. This message includes information about the infected machine: public IP address, hostname, username, etc. In addition to the heartbeat message, the malware starts the “command” loop that checks for new commands to execute every 3 seconds. If a command is received it executes it by spawning a PowerShell instance by executing “%windir% \SysWOW64\WindowsPowerShell\v1.0\powershell.exe” and returns the result to the C2 server.

```
service Implant { rpc FetchCommand(FetchCmdRequest) returns FetchCmdResponse {} rpc Heartbeat(HeartbeatRequest) returns Empty {} rpc Login(ImplantLoginRequest) returns ImplantLoginResponse {} rpc SendCmdOutput(SendCmdRequest) returns SendCmdResponse {} } message FetchCmdRequest { string id = 1; } message FetchCmdResponse { string adminId = 1; bytes cmdText = 2; } message HeartbeatRequest { string id = 1; bytes sysInfo = 2; } message ImplantLoginRequest { string id = 1; bytes sysInfo = 2; string password = 3; } message ImplantLoginResponse { string token = 1; } message SendCmdRequest { string adminId = 1; bytes output = 2; bytes error = 3; } message SendCmdResponse {} message Empty {}
```

## Client Component (GraphSteel)

The “client” component is a credential and file stealer. It communicates with the C2 server over WebSockets to a GraphQL endpoint. All the messages are encrypted with AES. The key is received from the C2 server. The malware author has written their own RSA implementation for the key exchange that is used to receive the shared secret. All messages prior to the key exchange, including the key exchange itself, are encrypted with AES using a hardcoded key.

The credentials on the machine are stolen using code lifted from goLazagne. In addition to stealing credentials, the malware will look in the user’s “Documents”, “Downloads”, “Pictures”, and “Desktop” folder for files with the file-extensions listed in the snippet below.

```
.key, .crt, .json, .csv, .7z, .rar, .zip, .ssh, .ovpn, .pptx, .xlsx, .docx, .ppt, .xls, .doc, .txt
```

If it finds a file with a matching file extension, it generates the MD5 hash for the file and checks with the C2 server if the file has already been uploaded. If it hasn’t, the file is uploaded to the C2 server.

## Infrastructure

Using the embedded CA certificate in the malware, we uncovered older service certificates and IP addresses. The oldest certificate we discovered had a validity date of “not before” December 9th, 2021. The IP address of the server where this certificate was served from is owned by the Russian hosting provider Zservers and still returned Elephant components as of April 2022. The components retrieved from the server are very similar to the samples used at the end of March. The other hosting providers used by the threat actor are PQ Hosting and Serverion.

> **Zservers.RU - Мы работаем только для вас!**
> Есть вопросы? Ответим на них 24x7; Telegram: @**zservers**; Jabber: **zservers**@exploit.im;
> ICQ: 631705; Email: support@**zservers**.ru; Авторизация.


> **Zservers.RU - Мы работаем только для вас!**
> Есть вопросы? Ответим на них 24x7; Telegram: @**zservers**; Jabber: **zservers**@exploit.im;
> ICQ: 631705; Email: support@**zservers**.ru; Авторизация.


The threat actor has chosen to sprinkle some French connections within their infrastructure. The certificates use French location information and the domain name forkscenter[.]fr. We don’t know if the decision to name it Elephant is a nod to Babar, an espionage software attributed to a French intelligence agency, or intended as a “false flag.”

## Conclusion

UAC-0056 (TA471, SaintBear, UNC2589) started using a new malware framework called Elephant back in December 2021. The malware has been delivered in targeted spear phishing campaigns using spoofed Ukrainian governmental email addresses. The malware consists of at least four different components that are used for stealing credentials, documents, and to provide remote access to the infected machine. The threat actor has opted to use multiple protocols for C2 communication, gRPC and GraphQL over websockets. This is an interesting choice as it complicates the development of the framework with more code to maintain.

## Indicators of Compromise

194.31.98.124 - C2 address used by the malware 87.249.139.161 - IP address that was used for sending the emails IP addresses used in previous campaigns. 91.242.229.35 45.84.0.116 80.66.76.187 Original Email 3c2022fea48b52326f9eec4c1c84f10b xls da305627acf63792acb02afaf83d94d1 Base-update.exe 06124da5b4d6ef31dbfd7a6094fc52a6 Java-sdk 36ff9ec87c458d6d76b2afbd5120dfae oracle-java 4a5de4784a6005aa8a19fb0889f1947a Microsoft-cortana 6b413beb61e46241481f556bb5cdb69c Samples from oldest server found 8e0eb1742b47745ff73389673996e964 cbc0e802b7134e1d02df1f2eb1b1d1e2 628f41776ae3b2e8343eeb9cdcd019f2 More Emails linked to the same sender IP C7051e88ae43c1bd4b869cf18280ec5e 40b42005e9cfc5ea2a7cfc1ced975cbb Fbe3cb4dfce740c3728c459b853e4249 (email submitted and target Romania)

Paths %HOME%\AppData\Local\Temp\Base-Update.exe %HOME%\.java-sdk 
Joakim Kennedy

Dr. Joakim Kennedy is a Security Researcher analyzing malware and tracking threat actors on a daily basis. For the last few years, Joakim has been researching malware written in Go. To make the analysis easier he has written the Go Reverse Engineering Toolkit (github.com/goretk), an open-source toolkit for analysis of Go binaries.


Nicole Fishbein

Nicole is a malware analyst and reverse engineer. Prior to Intezer she was an embedded researcher in the Israel Defense Forces (IDF) Intelligence Corps.

elephant Golang IoCs Phishing Research Ukraine