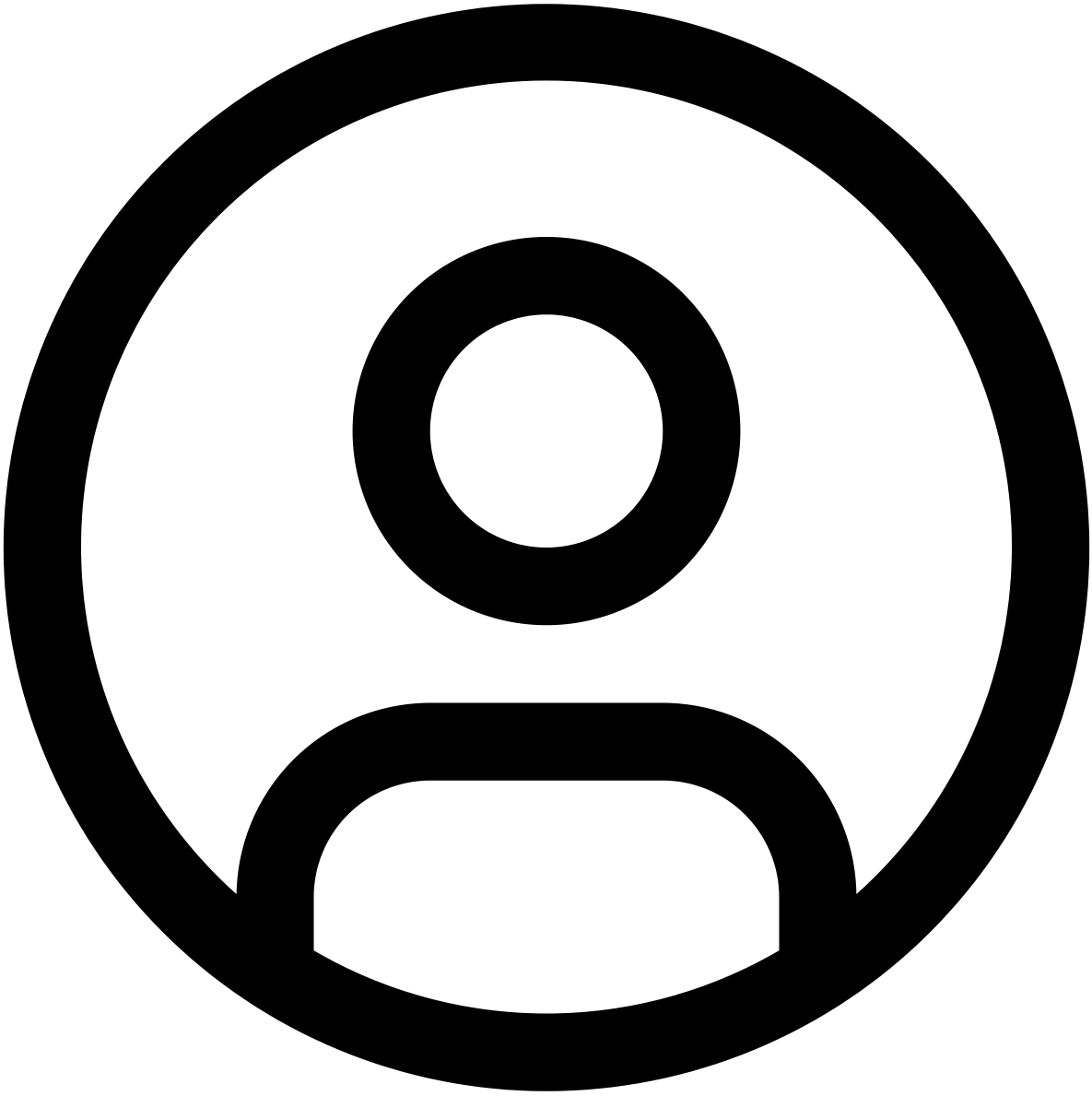
IBM X-Force discovers new Sheriff Backdoor used to target Ukraine

ibm.com/think/news/x-force-discovers-new-sheriff-backdoor-target-ukraine

Author



Golo Mühr
Malware Reverse Engineer

IBM

IBM X-Force discovered a set of previously unknown malware (Sheriff backdoor) used in a cyber espionage attack against an entity within Ukraine's defense sector in the first half of 2024. The threat actor used a popular news portal in Ukraine, ukr.net, to host the Sheriff backdoor. The modular backdoor can execute actor directed commands, collect screenshots, and covertly exfiltrate victim data using the Dropbox cloud storage API. During the investigation, X-Force identified threat activity similar to CloudWizard APT and Turla (aka ITG12), both of which are Russia-nexus threat groups that targeted Ukrainian entities.

Key findings

- A novel loader and modular backdoor were used in an attack in the first half of 2024 targeting the defense sector of Ukraine.
- The Sheriff Downloader module downloads a payload from **ukr.net**, a popular Ukrainian news portal, suggesting the website may have been compromised to facilitate the attack
- The backdoor uses the Dropbox API for command and control (C2) communication to facilitate encrypted data exfiltration

- Sheriff is able to download and manage multiple modules, including a screenshot module that was discovered during the investigation; commands and configuration values are received as ZIP file comments.
- X-Force identified similarities with CloudWizard APT and Turla (ITG12), Russian-nexus threat groups that target Ukraine entities.

Technical overview

During the investigation, X-Force reconstructed the following infection chain:

Deputy loader

The analyzed sample is a x86 DLL, containing a single export "MyFunc". It contains a hardcoded relative path to the Sheriff Downloader Module DLL in a Russian-language resource with the name "LoaderPath". The hardcoded path is

Local\WPnqv0hW\DZtdI.dll
which is concatenated with the following path: "%USERPROFILE%\AppData\".

The DLL then uses the *CLRCreateInstance* API to host the .NET Common Language Runtime (CLR). The Sheriff Downloader Module DLL's function **Loader.MainCycle.Run** is executed via the *ExecuteInDefaultAppDomain* method.

This technique was recently detailed in a blog post by X-Force as a red team technique.

Deputy Loader also supplies the paths of both DLLs as arguments separated by a semicolon (";").

Sheriff Downloader module

The Sheriff Downloader module is a x86 .NET DLL, which contains a library (dotnetzip.dll) packed into a resource using Fody Costura.

The class "MainCycle" contains the main function "Run", which begins by retrieving four values stored within Russian-language resources of the binary:

Using these, the sample attempts to download a file from

http://ukr[.]net/8V3fDJ0U/RDZXVh
into the folder

%APPDATA%\Xpgx2dAn\RDZXVh
If the file already exists, the download behavior is skipped, and the malware acts as a loader only.

Next, it decrypts the payload via the custom implemented "SymmetricCrypt" library and the password "BS7imxwRXueassn". The algorithm appears to be identical to the .NET built-in AES encryption (https://gist.github.com/jbtule/4336842#file-aesthenhmac-cs).

The resulting ZIP file is extracted in memory revealing at least two files which are sorted by file size. Finally, the first file (Sheriff Main Module) is reflectively loaded as a .NET assembly, calling the "MainClass.Run" method. The last file (Sheriff Init File) is read line-by-line and supplied to the executing assembly as a list object argument, together with the paths of the Deputy Loader DLL, the Sheriff Downloader DLL and the downloaded payload "RDZXVh".

Potential compromise of ukr.net

The download URL immediately raises concerns, as the host **ukr.net** is ranked as the 4th most visited website in Ukraine according to Semrush. **Ukr.net** is also an Internet Service Provider (ISP), a popular email provider, and hosts one of the largest news portals in Ukraine, with more than 100 million visits per month. Although **ukr.net** does appear to provide hosting services as well, it is generally not possible for users to host files on arbitrary root directories on the main web server. Therefore, it is likely that the threat actor compromised **ukr.net** in order to stage the encrypted Sheriff backdoor payload in early March 2024.

At the time of the investigation, the payload was not available, and X-Force was not able to identify other malicious payloads hosted on **ukr.net.** It is possible that the threat actor's access was limited in scope, only available for a short time, or intentionally used sparingly. A threat actor's access to Ukraine's largest news portal would position them to conduct a range of high-impact attacks and operate with enhanced obfuscation. In this specific incident, the threat actor may have abused the trusted domain to stage malware without raising suspicion.

Sheriff Main Module

The Sheriff Main Module is a x86 .NET DLL, which again contains a library (dotnetzip.dll) packed into a resource using Costura.

Initialization and configuration

When first executed, the "Run" function in the main class begins by reading the arguments received as a list from the Sheriff initialization file. It assigns the following values:

_symKey: "tkE7BqJ45HKwOes"
ConfName: "mInv.cfg"
ModulsFolder: "DxyVS1"
UploadLocalFolder: "gyTufW"
_defaultZipExt: "d7r"
refreshToken:
"sPfSlLkE3UcAAAAAAAAAAeUXe9lToajHac8y3w_9mmDptZKSU_Q0wdd4XSCZxfaU:yfw5e008wxkqbxg:ax6a2el8rf4cjo0"
_guid: "W5d2090860fd54c17809fb4da0b42b34d.test"
_asymPrivKey:
"MIICXAIBAAKBgQCuqWqUX60ArocW6V8zJN0vZ0CRAiY2jL+Ohjunh3p7wgac57Lwrmj0NIK80eLAO1zBIWEJZHH8vgapuLbv857SdG0Yw9iyGT
_asymPubKey:
"MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDBT+7FB360l8kvgAmCUKYjWxcybjTJiq81x0s56g+rxdvxHOlayuYhSb7xJ2JKv/1k1LtnAT

The table below lists an explanation for each of the arguments:

Name	Explanation
_symKey	AES key used to decrypt the config
ConfName	Filename of the configuration file
ModulsFolder	Name of the folder used to download further modules
UploadLocalFolder	Name of the folder used to exfiltrate data
_defaultZipExt	Default extension used to identify ZIP files
refreshToken	OAuth refresh token for authentication with the Dropbox API
_guid	String used together with Serial Number as victim ID
_asymPrivKey	RSA private key used to decrypt downloaded modules
_asymPubKey	RSA public key used to encrypt data before exfiltration

The asymmetric keys are from two different sets of keys, which prevented the decryption of exfiltration data during the investigation.

Next, Sheriff creates the local upload and download folders. If not present, the configuration file **mlvn.cfg** is written when the Sheriff Main Module is first executed. After that, it can be read and modified to maintain separate configs for each module. The decrypted config file contains the following values for the "main" module, separated by a semicolon (";"):

1. refreshToken
2. Maximum ZIP file size for exfiltrated data
3. Maximum file size for exfiltrated data before compression
4. URL used to retrieve the victim's public IP address
5. Minimum interval (ms) used to download and upload files
6. Maximum interval (ms) used to download and upload files

[main];sPfSlLkE3UcAAAAAAAAAAeUXe9lToajHac8y3w_9mmDptZKSU_Q0wdd4XSCZxfaU:yfw5e008wxkqbxg:ax6a2el8rf4cjo0;220200960;20971
[key];1024

No "key" module was found during the investigation. This module might have been responsible for establishing persistence for the Deputy Loader DLL, which writes the following registry key:

HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\MicrosoftEdgeAutoLaunch_<REDACTED>

The key contains the command to execute the Deputy Loader's MyFunc export:

"C:\WINDOWS\SysWOW64\rundll32.exe" "<LOCAL_APPDATA>\GRPRhuI4\T5CBy.dll",MyFunc

Dropbox C2

Command and control communication is managed through a class "DbApiV2", which uses the Dropbox API to create, find, download, upload, move and parse remote files and folders on Dropbox. It uses the refresh token to get a temporary access token for authentication via the following URL:

https://api.dropbox.com/oauth2/token

These API endpoints are used to manage remote files and folders:

https://content.dropboxapi.com/2/files/upload
https://content.dropboxapi.com/2/files/upload_session/start
https://content.dropboxapi.com/2/files/upload_session/finish
https://content.dropboxapi.com/2/files/upload_session/append_v2
https://content.dropboxapi.com/2/files/download
https://api.dropboxapi.com/2/files/delete
https://api.dropboxapi.com/2/files/create_folder
https://api.dropboxapi.com/2/files/list_folder

Before attempting to download files, the Sheriff Main Module uploads a log message containing the public IP address of the victim and the list of loaded modules. The log is XOR-encrypted using the victim ID consisting of the GUID (from the arguments or randomly generated) and the serial number. After encryption, the log is uploaded to a Dropbox folder named to match the victim ID.

All files are retrieved from the Dropbox folder at **/<victim_id>/Dow/** and are downloaded to the local "ModulsFolder" hardcoded as "/DxyVS1". After download, all files are instantly deleted from Dropbox. Next, we will discuss how the downloaded files are handled by the main module.

The upload process begins by enumerating all local files in the "UploadLocalFolder", in this case hardcoded as "/gyTufW". Depending on their extensions, they are sorted into three categories:

1. Files using the hardcoded "_defaultZipExt" **.d7r** are already zipped;
2. Files without an extension are already encrypted and ready for upload; and,
3. All other files are still in clear text.

The function "PreparingForUpload" will then compress all clear text files into a new ZIP file. All ZIP files are subsequently encrypted using a randomly generated AES key which is in turn encrypted using the public RSA key and concatenated with the encrypted file. During execution, the function deletes all residual files from the folder until only fully compressed and encrypted files are left. These are then uploaded to the Dropbox folder at **/<victim_id>/Up/** while deleted locally.

Both the upload and download functions are executed asynchronously and run with a timer hardcoded to 30 seconds in the analyzed sample.

At the time of investigation, the *Dropbox* account did not host any files anymore, as indicated by the space usage:


```
{
  "used": 0,
  "allocation": {
    ".tag": "individual",
    "allocated": 2147483648
  }
}
```


The associated *Dropbox* account displays the following information:


```
{
  "account_id": "dbid:AABLMHYTVufS0NLF_cnID1nm_-R9m1aj9ds",
  "name": {
    "given_name": "Poco",
    "surname": "Poco",
    "familiar_name": "Poco",
    "display_name": "Poco Poco",
    "abbreviated_name": "PP"
  },
```

```
  "email": "poco.m5.miui.13@gmail.com",
  "email_verified": true,
  "disabled": false,
  "country": "DE",
  "locale": "ru",
  "referral_link": "https://www.dropbox.com/referrals/AABf13Qzpq31wbsquudl7xfwagHIa8GgdZg?src=app9-5618657",
  "is_paired": false,
  "account_type": {
    ".tag": "basic"
  },
  "root_info": {
    ".tag": "user",
    "root_namespace_id": "2199102147",
    "home_namespace_id": "2199102147"
  }
}
```


Sheriff modules and commands

The Sheriff main module's task is to act as an orchestrator to launch and manage different modules. These modules may be download via the process described above, one of which was discovered during the investigation ("./DxyVS1/dowtuxZml").

The "LoadModuls" function iterates through downloaded files, decrypting them using the RSA private key and the resulting AES key. The decrypted ZIP file contains a comment string, which is used to parse the module:

The comment is separated into several values via the pipe symbol ("|") and further into sub values separated by a semicolon (";").

D|scr;ScreenShot.Shot;LoadDll;KillDll;ConfDll|0|0;None

The following is a description of the values after parsing:

1. Command: "D"
2. ModInfo (part of an object maintaining information about each module during execution of the main module)
    1. Marker: "scr"
    2. NameSpace: "ScreenShot.Shot"
    3. LoadMethod: "LoadDll"
    4. KillMethod: "KillDll"
    5. ConfigMethod: "ConfDll"
3. On/Off (if the original file should be deleted)
4. NeedZip (if the module requires it's results to be zipped by the main module); Compression (the desired compression method, default is Deflate)

Sheriff accepts the following commands:

Command and description

- D — Parses the ModInfo from the comment and depending on the On/Off value deletes the original file after loading the .NET assembly. To load or kill the module (if already loaded) it uses the "LoadMethod" and "KillMethod" values respectively. The "LoadMethod" takes the "Marker", the "UploadLocalFolder" and the "_defaultZipExt" as arguments.
- T — Parses the ModInfo from the comment, deletes the original file and executes the "LoadMethod". The "LoadMethod" takes the "Marker", the "UploadLocalFolder", the "LoadMethod", the "KillMethod" and the "ConfigMethod" as arguments.
- E — Only accepts a single value (command argument) after the first pipe symbol ("|") and runs the function "RunExeInMemory". The file within the ZIP archive is dropped to a temporary path as %TMP%\<number_of_ticks>.exe and executed as a new process with the argument if supplied. The dropped file and ZIP file are subsequently deleted.
- C — Parses a text file within the ZIP, line by line, which can contain a list of custom commands (see second table below).
- R — Updates the main module by renaming the original file and writing the downloaded payload to the original path. Depending on the success, one of two status messages will be uploaded: "MainModule was successfully changed" or "MainModule was not changed".

The second table details a list of commands that can be read as a text file using the "C" command:

Command pattern	Description

(tree) <path_1> <path_2> ...	Uploads files from a list of specified paths.
(treedel) <path_1> <path_2> ...	Deletes files from a list of specified paths and uploads a log message "Files were deleted: <number_of_files>"
(cmd);value1;value2;...	Executes each value as a separate command in a new process "cmd.exe /c <value>", reads stdout and stderr, and uploads it as an RSA encrypted file to Dropbox.
[modname];value1;value2;...	Inserts the full string into the configuration file. Note, the "modname" is identical to a module's marker.
{modname};value1;value2;...	If the "modname" is "Suicide", Sheriff will kill all modules, delete all files and run a clean-up script. If the "modname" matches a loaded module, it will invoke the "KillMethod" and delete its corresponding file.

After all modules are loaded, the main module's "Run" function will iterate through each loaded module and invoke the "ConfigMethod", supplying the corresponding module's settings as parsed from the original config file. This likely allows operators to easily update several module's configs while they are running.

Screenshot module

One of the modules retrieved during the investigation is the Screenshot module. When the module is loaded, it receives the following arguments from the main module:

1. marker
2. uploadPath
3. defaultZip

The module still contains a default value "tgr" for "defaultZip", which is overwritten at that point. Using the module's "ConfigMethod", the main module is also able to specify the following configuration values:

1. ImageCount (number of screenshots taken until they are zipped). Default = 25
2. TimerCount (timer interval, how often a screenshot can potentially be taken). Default = 5 seconds
3. DefiniteShot (maximum time until a screenshot is definitely taken). Default = 15 minutes
4. BmpQuality (screenshot quality/compression parameter between 0-100). Default = 25
5. WindowsTitle (comma separated list of window titles which are specifically targeted for screenshots).

Once started, the module will check every 5 seconds (TimerCount) if it can take a screenshot. To take a screenshot, one of the following conditions has to be true:

1. The last screenshot has been taken more than 15 minutes prior (DefiniteShot);
2. The foreground window is different than the last time a screenshot was taken; or,
3. The title of the foreground windows matches any of the strings in the list "WindowsTitle".

During each shot, if the number of shots reaches "ImageCount", the existing screenshots are added to a ZIP file formatted as **{0:yyyy.MM.dd_HH.mm.ss}.jpg** using the screenshot's "DateTime" object.

The ZIP file name is formatted as **{0:yyyy.MM.dd_HH.mm.ss.ffff}.<defaultZip>** using the "DateTime" object at which the ZIP is created. The ZIP file also gets a comment consisting of the module's marker ("scr") as shown in the screenshot below.

Suicide function

Sheriff's main module also contains a Suicide function, which can be remotely invoked. The function stops all download and upload activity and then iterates through each module to invoke the corresponding "KillMethod". It then proceeds to delete the whole directory containing the main module and the global folder on *Dropbox* used for C2 communication. Next, the function searches for the path of the first stage loader (Deputy Loader) DLL within registry subkeys beneath:

HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Any subkeys containing the path are subsequently deleted.

Finally, Sheriff inserts the paths of the "loader" (Sheriff Downloader Module) and "loadDll" (Deputy Loader) into the following BAT file, drops it to %TMP% and executes it:

```
SET loader="<loader.path>"
SET loadDll="<loadDll.path>"
:loop
IF EXIST %loader% del /F %loader%
IF EXIST %loader% goto loop
:loop1
IF EXIST %loadDll% del /F %loadDll%
IF EXIST %loadDll% goto loop1
rmdir /s /q "<loader.directory_name>"
rmdir /s /q "<loadDll.directory_name>"
(goto) 2>nul & del "%~f0"
```

The script above will delete the files of both the Sheriff Downloader Module and the Deputy Loader as well as their respective directories, before deleting itself.

Similar threat actor techniques

During the analysis, several initial indicators point toward Russia-based threat actors, including:

- Deputy Loader and Sheriff Downloader contain Russian language resources;
- Russian locale used by the Dropbox account; and,
- Targeting Ukraine.

X-Force assesses the Sheriff backdoor is most likely a tool designed for cyber espionage and intelligence gathering versus financially motivated cybercrime. The malware focuses on exfiltrating data and taking screenshots while maintaining a low profile designed for prolonged compromises. It was developed with a clear intention of staying as covert as possible, ensuring communication and most artifacts dropped to the disk remain encrypted. Network communication remains stealthy through the abuse of the legitimate Dropbox API as well as **ukr.net**, a popular website in Ukraine, which is used to stage the malware. Sheriff also implements several self-destructive functions to delete traces after execution. Finally, the well-structured code, folder structure, modular implementation, logging and comprehensive functionality and configurability indicate an increased level of sophistication as would be expected from a state-sponsored group.

The investigation also revealed several minor overlaps with previously documented campaigns attributed to known Russian-nexus threat actor group Turla (aka ITG12). For example, the group's Kazuar .NET backdoor displays several similarities to Sheriff, including:

- Kazuar also maintains a slightly similar folder structure;
- Although different, Kazuar also generates a GUID and uses the victim's serial number;
- Kazuar also implements logging and uses AES and RSA encryption;
- Kazuar is also modular, although it appears to refer to "plugins" instead of modules;
- Kazuar also uses max and min interval values; and,
- Kazuar supports similar commands as Sheriff, including "Suicide", screenshot, command-line execution, binary execution, file deletion, exfiltration and self-updating.

Noticeably, the Crutch backdoor attributed to Turla by ESET also uses the Dropbox API for C2 communication in a similar way as Sheriff, although it is not .NET-based.

Further research also revealed Sheriff's overlaps with Operation Groundbait's Prikormka backdoor, including:

- Modular backdoor with a downloader, core and screenshot module;
- Prikormka also maintains two folders in %USERPROFILE%\AppData\Local\ for uploads and downloads;
- Prikormka also uses custom extensions to identify files meant to be encrypted and compressed before exfiltration;
- Prikormka's screenshot module used ".tgz" as part of the custom extension, Sheriff's Screenshot module uses ".tgr"; and
- Prikormka modules list "Cycle" as one of the required export functions, which is similar to the "MainCycle" class used by the Sheriff Downloader Module.

Kaspersky Labs later documented strong overlaps between Prikormka and CloudWizard APT. X-Force also noticed several similarities between Sheriff and CloudWizard, including:

- Modular backdoor with a main module managing configuration and C2 for each module;
- CloudWizard also uses AES and RSA to encrypt/decrypt ZIPs before/after uploading and downloading;
- CloudWizard also supports Dropbox as a C2 mechanism with OAuth authentication;

- Both CloudWizard and Sheriff contain a function "GetSettings"/"get_Settings" to retrieve each module's configuration;
- Both CloudWizard's and Sheriff's Screenshot module's support a "WindowsTitle" argument, to compare against the current window's title before taking a screenshot;
- CloudWizard, Prikormka and Sheriff share the same screenshot taking interval of 15 minutes; and,
- CloudWizard and Prikormka's file listing modules are called "tree", which is the name Sheriff uses for exfiltration of a list of files.

X-Force believes the Sheriff backdoor was used as part of a targeted operation. The malware is possibly related to Russia-aligned CloudWizard APT, which has been known to target entities in Ukraine in the past. There is a lower possibility of a connection to the Turla (ITG12) threat cluster due to minor overlaps in TTPs and malware.

Conclusion

The Sheriff backdoor and its use within cyber espionage operations detailed in this report displays several interesting features. First, the Sheriff backdoor is a well-designed modular espionage tool enabling long-term access to the victim's environment. Second, its modular structure and self-destroy features highlight the developers' concern of detection and analysis of their tooling. Next, the ability to stage the malware on **ukr.net** is also indicative of the advanced capabilities of the threat actor.

Recommendations

X-Force recommends individuals and entities associated with the government, military or defense sector of Ukraine remain in a heightened state of defensive security and to:

- Hunt for unusual traffic with public IP resolving services such as **https://api.ipify[.]org**
- Hunt for unusual traffic communicating with the Dropbox API:
  - https://api.dropboxapi.com
  - https://content.dropboxapi.com
- Install and configure endpoint security software.
- Update relevant network security monitoring rules.

Technical appendix

To make attribution more transparent and encourage more collaboration among researchers, the samples were uploaded to VirusTotal by IBM X-Force.

Indicator	Indicator Type	Context
60f20be29cafea3402c8cb396c1cb43ef21ec1b401ad1d4239c0a990670daa8d	SHA25	Encrypted Main Module "RDZXVh"
86b8d48df5787d57836276219a9e3dbc0d7e56d68cf99b514aca55564f818182	SHA256	Sheriff Initialization File "n5K3B"
8832fb7ef434a56f9d151d8e1ebda94544a90a420fee0820b5b08d95224763f5	SHA256	Deputy Loader "t5cby.dll"
8c22326d08a6334181c06e25c6df35032cd6916cfbe692d66fc8db3aa8b70e42	SHA256	Encrypted Screenshot Module "dowtuxZml"
8d4df90f4e7fc6d9d08d4b5a272037ee7c565def9df180ad1eb08efe8d357bd4	SHA256	Sheriff Main Module "1Pr3v"
92b9ef4e81610487ea9df255fa83a8e6c3bd2726ccdb909988e8c8b919506289	SHA256	Sheriff Configuration File "mInv.cfg"
e2b892533bd4135004778783b95e833fca6ee740bf0a1cb2d5d1a44b93fd7962	SHA256	Decrypted Screenshot Module ZIP file
ec84ae8db92a88109bc68baefc3b0a9de8579129d7a5a431072f09fdcb8c7862	SHA256	Sheriff Screenshot Module "NeXSv"
f9e237a939b998fe071e0101904f7d10cde6ce7b1cb4df1e7d345094af6b048e	SHA256	Sheriff Downloader Module "DZtdI.dll"
http://ukr[.]net/8V3fDJ0U/RDZXV	URL	Sheriff Download URL. Note, ukr.net is a legitimate website.

https://api.ipify[.]org	URL	Legitimate service used to determine public IP address, frequently abused by malware authors.

Mixture of Experts | 25 July, episode 65

Decoding AI: Weekly News Roundup

Join our world-class panel of engineers, researchers, product leaders and more as they cut through the AI noise to bring you the latest in AI news and insights.

Watch the latest podcast episodes →

Report Cost of a data breach report 2024
Data breach costs have hit a new high. Get essential insights to help your security and IT teams better manage risk and limit potential losses.

Read the report