6 DECEMBER 2022 • SALIM BITAM • DANIEL STEPANIC • SETH GOODWIN • ANDREW PEASE

Exploring the REF2731 Intrusion Set

REF2731 intrusion set, campaigns, and malware observations

42 min read   Campaigns, Attack pattern, Malware analysis

Key Takeaways

- PARALLAX loader maldoc campaigns continue to have success delivering the NETWIRE RAT.
- The PARALLAX loader leverages advanced features including DLL-side loading, syscall usage, process, and steganography.
- Shared infrastructure can be used to stitch campaigns and intrusion sets together.

Preamble

which deploys the NETWIRE RAT. This activity has managed to stay under the radar with low detection rates and continues to incorporate interesting techniques such as DLL side-loading, syscall adoption, process injection, and leveraging steganography.

PARALLAX is a full-featured modal backdoor and loader featuring defense evasion and information on stealing capabilities, first observed in 2020 and associated with COVID-19 malspam campaigns. NETWIRE is a mature and cross-platform RAT that was first observed in 2012

In this research publication, we will go through the execution flow of one of the observed campaigns, the different features of the PARALLAX loader, technical analysis around the campaigns, campaign intersections, detection logic, and atomic indicators.

Execution Flow (PARALLAX loader)

The Elastic Security Labs team has been monitoring multiple campaigns over the past year leveraging the PARALLAX loader. PARALLAX has multiple capabilities and use cases. This analysis observed the PARALLAX loader being used to load other remote access tools (the NETWIRE RAT). Using our PARALLAX payload extractor, we have also observed the PARALLAX loader being used to load the PARALLAX RAT for interactive remote access. These infections typically start through email spam campaigns delivering macro-enabled lure documents.

> *"On July 27, 2022, Microsoft began rolling out a change to Office documents that will prevent users from opening macros in files that came from the Internet, such as email attachments. We have not observed a change in TTPs based on this update from this intrusion set. Our sampling for this research of macro-enabled Word documents started in March of 2022 and continued through August 2022."*

High-level summary of the execution flow:

1. An email is sent to a victim with a macro-enabled Microsoft Word document attachment.

3. The Microsoft developer tool (**MsiDb.exe**) sideloads the malicious (**msi.dll**).
4. This malicious DLL drops and decrypts a WAV file (**cs16.wav**) before injecting the contents (shellcode) into **cmd.exe**.
5. The injected shellcode is used to extract the NETWIRE RAT and set up the PARALLAX loader from a dropped image (**paper.png**) and inject into **cmd.exe.**
6. A scheduled task is used to establish persistence for the PARALLAX RAT.
7. The NETWIRE payload is then executed and sets up its own persistence mechanism.

First Stage (lure/macro)

The first stage in these campaigns involves macro-enabled lure documents typically with themes around United States tax filings.

In this lure, we observed legitimate code lifted from the GLPK (GNU Linear Programming Kit) used to bypass static analysis of the macro. The malicious code is then interwoven within the macro making it look very genuine and more deceptive.

This approach to obfuscation is also observed when critical components used for the next stage are not stored in the macro itself but called from text buried several pages deep within the lure document.

The macro parses the embedded paragraph text on page three of the lure document and locates the object names and next stage components based on their string length. This is a clever technique to avoid detection based on static analysis of the macro (green text comments added to the images below by ESL for clarity).

The macro then uses the **CreateObject** function to create the required objects and download each of the malware components, saving them to the **AppData** directory of the current user.

It then executes **AppData\MsiDb.exe** through the created **wscript.shell** object.

For this observed lure, the five components that are downloaded for the next stage as identified in the embedded text image above are:

File name	Description
MsiDb.exe	Legitimate Microsoft development application used to import/export database tables and streams
msi.dll	Malicious DLL used for side-loading
cs16.wav	XOR encrypted shellcode
paper.png	Obfuscated NETWIRE and additional PARALLAX loader stager
cs16.cfg	Configuration containing the location of the next execution stage png file, it can either be local or hosted in a remote server

Second Stage (MsiDb.exe)

One of the key strengths in these campaigns is its ability to bypass static detection by modifying legitimate DLLs, a common trend previously reported with the BLISTER loader analysis [1, 2]. Once all the components are retrieved, the macro executes the signed Microsoft development tool (**MsiDb.exe**) to load the previously downloaded malicious library (**msi.dll**).

When the campaign began in September of 2022, this DLL had zero detections in VirusTotal due to its DLL tampering technique where a slight modification of a benign function is overwritten with the second stage.

When (**MsiDb.exe**) sideloads the malicious (**msi.dll**) module, we can see the difference between the patched and unpatched version of **msi.dll**.

During this loading stage, the malicious code is heavily obfuscated and leverages **dynamic API resolution** to bypass static analysis tools and processes. It performs this using two functions:

- One function is used to retrieve library addresses using the CRC32 checksum hash of the requested library name.

The malware then builds its own import table, storing it on the stack. An interesting aspect is that the malicious code performs an anti-analysis check to see if the current process name matches the targeted application (**MsiDb.exe**), if it doesn't match, the malware will stop at this stage. This check will hinder automated dynamic analysis systems that might try to analyze **msi.dll** in isolation by executing it with other common applications such as **rundll32.exe** or **regsvr32.exe**.

Next, the malware will load **cs16.wav** and XOR-decrypt it using a key embedded in the file. The key resides in the 200 bytes following the first 4 bytes of the file (bytes 5-204).

The malware will then execute the shellcode inside the decrypted WAV file.

Third Stage (shellcode)

To evade user mode hooks utilized by EDR/AV products and as debugger breakpoints, the malware uses direct system calls to low-level APIs used for process injection. It performs this by first mapping a file view of the Windows **ntdll.dll** library from the System directory.

It then retrieves the API offset by subtracting the API address from the loaded base address of the loaded **ntdll.dll** , then finally it will use the offset from the mapped **ntdll.dll** and extract the syscall number.

After this, the loader uses the Heaven's Gate technique and performs injection in the suspended **cmd.exe** process leveraging native Windows **ZwAllocateVirtualMemory** , **ZwWriteVirtualMemory,** and **ZwResumeThread** API functions.

Fourth Stage

One interesting technique observed during this stage is through the use of a dropped file (**cs16.cfg**). The file is a legitimate Python header file and is prepended with the next stage file name (**paper.png**). In our observations, these point to local files previously downloaded but also has the flexibility to point to hosted objects. This is another example of using benign code to obfuscate more malicious intent.

Object Model (COM) interface to download a PNG file and store it on disk (**paper.png** in our example).

The malware extracts a configuration structure from the stenographically-obfuscated PNG that contains the next PARALLAX loader stage and the final payload; in our sample, we identified the final payload as the NETWIRE RAT, but this process could be used to deliver other payloads.

The malware executes position independent shellcode that reads and decodes the PNG file, it first extracts the red pixel bytes to an array by parsing the PNG, then decompresses the data with the **LZMA algorithm**.

Next, it creates a suspended **cmd.exe** process and injects the NETWIRE payload and the last PARALLAX stage that will set up the environment and execute the NETWIRE payload.

Below is the memory regions showing the injected process hosting the NETWIRE payload:

Fifth Stage

The fifth and final stage of PARALLAX Loader performs a UAC bypass through **CMSTPLUA** COM interface, a technique that has been used by ransomware-like LockBit, it then sets persistence on the system before executing the final payload by creating a scheduled task to run **Msidb.exe** using Component Object Model (COM).

Campaign Analysis

Throughout the analysis of the lure documents and malware families, we observed two campaigns associated with their TTPs, malware, network infrastructure, and lure metadata.

The intersections we observed allowed us to observe additional network infrastructure and identify the characteristics of one infrastructure owner in Campaign 1.

well as their relationships.

This section will be focused on campaign intersections. As each campaign functioned similarly with respect to their technical implementation (lure document -\> macro -\> defense evasion techniques -\> PARALLAX loader -\> NETWIRE RAT), we'll use the analysis of the five stages for the deployment of the PARALLAX and NETWIRE malware that has been described in detail in the previous Execution Flow section.

While we are not attributing these campaigns to any specific threat actor, we have identified parallel research leveraging the same TTPs that we observed. This research was attributed to the financially motivated threat group, Evilnum [1, 2] and the DarkCasino campaign.

Campaign 1

Overview

This campaign is clustered by shared lure document metadata, network infrastructure, dropped macro, and malicious DLL (**msi.dll**) .

Lure Documents

The three lure documents used in Campaign 1 were all macro-embedded Microsoft Word documents. The documents were all 153 pages long, with the macro embedded on the 3rd page. The documents all included the H1 Word document header of **Как я искал Гантмахера** (loosely translated to: "How I searched for Gantmakher"). Vsevolod Gantmakher was a Russian physicist.

Extracting the metadata for all three documents, we can see their relationships based on several fields; most notably:

- The identical **HeadingPairs** (the names of the Word document header).
- The identical **CreationDate** dates.
- The identical **LastPrinted** dates.

The H1 document header of the lure documents does not appear relevant to the targeting as the lure document names and lure document content are wholly unrelated: two of the three document names were related to 2021 United States tax filings, all three of the document names are in English, and the contents of the lure documents are in Cyrillic.

Macro

The macro downloads five files, detailed in the Execution Flow section above (**cs16.wav**, **msi.dll** , **MsiDb.exe** , **paper.png** , and **cs16.cfg**), from a different domain for each lure document.

Network Infrastructure

Campaign 1 included three domains contacted by the macro to download artifacts required for stages two through five (described in the "Execution Flow" section above) and three domains used for the NETWIRE RAT C2.

The six domains are:

- digitialrotprevention[.]com - macro-connected.
- internationalmusicservices[.]com - macro-connected.
- globalartisticservices[.]com - macro-connected.
- ohioohioa[.]com - NETWIRE C2.
- ywiyr[.]com - NETWIRE C2.
- septton[.]com - NETWIRE C2.

The macro-connected domains (digitialrotprevention[.]com, internationalmusicservices[.]com, and globalartisticservices[.]com) include metadata that has allowed us to cluster these three domains together in Campaign 1.

In the above image, the Admin email address and Admin user name is russnet123@protonmail[.]com and **rus fam** , respectively. As of this writing, these domains have been suspended.

> *email address and Admin user name. The lure document included similar US tax document themes, macro elements, and TTPs; but we were unable to confirm that it was part of this campaign. This lure document was first observed in May of 2022 and is possibly part of a testing wave, but this is speculation. We are confident this is a malicious domain and are including it as an indicator artifact for this intrusion set, but not this campaign."*

Once the execution flow reaches the Fourth Stage (described in the Execution Flow section above), the final three domains (ohioohioa[.]com, ywiyr[.]com, and septton[.]com) act as ongoing command and control nodes for the NETWIRE RAT.

While ohioohioa[.]com and ywiyr[.]com are protected by privacy services, septton[.]com has interesting metadata that we were able to collect and is outlined below in the SEPTTON Domain section below.

Campaign 1 Indicators

Name	STIX 2.1 Indicator Type	Identifier
bc9f19ae835d975de9aaea7d233b6ea9b2bc30f80d192af2e8e68542b588917e	SHA-256	Brian_Tax_Docs.doc lure document
d70365481fb4806130743afd199697eb981a0eb2756754ecc548f5b30c2203a5	SHA-256	VIRGINIA-TAX-RETURN-2021-US-EXT.doc lure document
9dd709cb989d985a6cfee4a254f894a3b878a03962dbf253cb09a24ece455d58	SHA-256	All Docs.doc lure document
16227f50bbe42a13a2abf0bf0e146f356863de59525c54909ea8ccc2db448f77	SHA-256	msi.dll PARALLAX loader / NETWIRE

2a[illegible]7e49abdbb222983fa6044fdf66		
6ed65beb692301af5296ba6751063ae40e91c4e69ced43560c67ce58165c36b5	SHA-256	cs16.cfg (config for PNG stage)
5f259757741757c78bfb9dab2cd558aaa8403951c1495dc86735ca73c33d877f	SHA-256	paper.png (stager for NETWIRE)
globalartisticservices[.]com	domain-name	PARALLAX loader domain
DigitalRotPrevention[.]com	domain-name	PARALLAX loader domain
InternationalMusicServices[.]com	domain-name	PARALLAX loader domain
russnet123@protonmail[.]com	email-addr	PARALLAX loader domain registration email address
chisholm.i@aol[.]com	email-addr	NETWIRE C2 domain registration email address
ywiry[.]com	domain-name	NETWIRE C2 domain
ohioohioa[.]com	domain-name	NETWIRE C2 domain
septton[.]com	domain-name	NETWIRE C2 domain

Campaign 2

Overview

This campaign is clustered through its lure document metadata, network infrastructure, dropped macro, and malicious DLL (**msvcr100.dll**).

The lure document used in Campaign 2 is a macro-embedded Microsoft Word document. The document metadata differentiates it from Campaign 1 based on the **LastModifiedBy** field and the macro network infrastructure.

The document name was also related to 2021 United States tax filings.

Macro

Like Campaign 1, the macro downloads several files. Beyond the DLL file (**msvcr100.dll**), all files were offline before they could be collected. Based on the TTPs observed in this campaign, we assess with high confidence that they **(java.exe**, **Fruit.png** , **idea.cfg** , and **idea.mp3**) function similarly to the files from Campaign 1 and detailed in the Execution Flow section above.

Additional details about the Campaign 1 and Campaign 2 file relationships are in the "Campaign intersections" section below.

Network Infrastructure

Campaign 2 included one domain contacted by the macro to download artifacts required for stages two through five (described in detail in the "Execution Flow" section above). Additionally, there was one domain used for the NETWIRE RAT C2.

The two domains are:

- solro14.s3.ap-northeast-3.amazonaws[.]com - macro-connected
- ohioohioa[.]com - NETWIRE C2

Once the execution flow reaches stage four, ohioohioa[.]com acts as the ongoing command and control node for the NETWIRE RAT.

Campaign 2 Indicators

Name	STIX 2.1 Indicator Type	Identifier
solro14.s3.ap-northeast-3.amazonaws[.]com	domain-name	PARALLAX loader domain
32fc0d1ad678133c7ae456ecf66c3fcf97e43abc2fdfce3ad3dce66af4841f35	SHA-256	2021-Individual-Tax-Form.doc lure document
443879ee2cb3d572bb928d0831be0771c7120968e442bafe713a6e0f803e8cd9	SHA-256	msvcr100.dll PARALLAX loader / NETWIRE
ohioohioa[.]com	domain-name	NETWIRE C2 domain

Campaign Intersections

Campaign 1 and Campaign 2 intersect in several ways.

As illustrated in the image below, each campaign relied on a lure document (or documents) to execute a macro that contacted adversary-owned or controlled domains; downloaded artifacts used to install and protect the PARALLAX and NETWIRE RAT implants. Additionally, in both campaigns we analyzed, there is a shared network infrastructure used for the NETWIRE C2.

The Pyramid of Pain

In 2013 (and updated in 2014), security researcher David Bianco released an analytical model called the Pyramid of Pain. The model is intended to understand how uncovering different parts of an intrusion can impact a campaign. As you can see in the model below, the identification of hash values is useful, but easily changed by an adversary whereas identifying TTPs is very difficult for an adversary to change.

the impact (read: the amount of "pain") you can inflict.

When analyzing the two campaigns, we can put the Pyramid of Pain into action.

- **Hash values** - each lure document had a unique hash.
- **IP addresses** - each network connection leveraged a different IP address.
- **Domain names** - each network connection leveraged exclusive domains for the macro components but shared a NETWIRE C2 domain (ohioohioa[.]com).
- **Network/host artifacts**
  - Identically-named host artifacts observed in Campaign 1.
  - Renamed from Campaign 1, but functionally identical, host artifacts observed in Campaign 2.
  - Artifact bundles from both campaigns include similarly formatted and functionally identical files.
- **Tools** - macro-enabled Word document lures, and PARALLAX and NETWIRE RATs.
- **TTPs** - complex and defensive five-staged execution chain.

Looking across both campaigns, we can see there is some shared infrastructure at the Domain Names tier in the NETWIRE C2 domain (ohioohioa[.]com). In the Network/host artifacts tier we can see additional intersections between the campaigns.

In both campaigns, we can see a PE file (**MsiDb.exe** and **java.exe**), a DLL file (**msi.dll** and **msvcr100.dll**), a PNG file (**paper.png** and **Fruit.png**), an audio-format named file (**cs16.wav** and **idea.mp3**), and a configuration file (**cs16.cfg** and **idea.cfg**) at the Network/host artifact tier. All downloaded files in Campaign 1 are named the same across all three lure documents. In both campaigns, the audio-format named files have the same base name as the configuration files (**cs16.wav** / **cs16.cfg** and **idea.mp3** / **idea.cfg**). In both campaigns, we assess with high confidence that all host artifacts are functionally identical as described in the Execution Flow section above.

As reported in the Campaign 1 section, most of the network infrastructure was either well-used across multiple intrusions unrelated to our campaigns or protected by domain privacy services.

An exception to that is the seppton[.]com domain, which was used as the C2 node for a NETWIRE RAT implant in our sampling. Continuing to analyze this domain, we observed several other associated malicious files. While we did not independently verify the family of malware that is communicating with this domain, signature names in VirusTotal include NETWIRE.

> ***"It should be noted that signature names in VirusTotal alone do not present enough information to provide a high-confidence conviction of a malware sample to a malware family."***

Looking through the registration information for the domain, we observed two elements of note, both email addresses - marketforce666@yandex[.]com and chisholm.i@aol[.]com.

In the next two sections, we'll discuss the resource development for domains used in campaigns.

marketforce666

Searching for **marketforce666** in a search engine did not return results of value from the United States; however, when changing to an Internet egress point within Russia and using the Yandex search engine (Yandex is a Russian Internet services provider), we identified 802 results that show this term has been associated with multiple abuse reports.

When expanding our search for domains registered by marketforce666@yandex[.]com, we identified three additional domains. We did not observe these additional domains in our campaigns, but we are including them as indicator artifacts. Below are the four total domains (one from Campaign 1 and three additional) that were registered by, either as the admin, tech, or registrant address, marketforce666@yandex[.]com.

gaza666

a moniker of **gaza666** from the online forum and marketplace, Infected Zone.

On this forum, the user **gaza666** attempted to purchase (`https://infected-zone[.]com/threads/2814/`) an "Office 365 Complete Package" from the online seller **rzkyo**. **gaza666** and the seller **rzkyo** engaged in a dispute on the forum where **gaza666** did not believe they received what they purchased - which was a package for email spamming and four United States Office 365 accounts but received three nonfunctional and non-Office 365 Phillipino accounts. The seller, **rzkyo** , responded and the two debated what was purchased and what was delivered. The dispute was responded to by a moderator who attempted to resolve the issue.

The results of the dispute were not in the forum, but there were several screenshots where **rzkyo** showed **gaza666** and the moderators that the services they sold were functional.

While it is unknown if the infrastructure above that **gaza666** attempted to purchase from **rzkyo** was used in our observed campaigns (or ever used at all), but **gaza666** is associated with chisholm.i@aol[.]com, which was used to register septton[.]com, and septton[.]com was used as a NETWIRE C2 node in Campaign 1.

marketforce666 (marketforce666@yandex[.]com) and **gaza666** (chisholm.i@aol[.]com) share a relationship in that both emails were used in the registration of septton[.]com, which was used as a NETWIRE C2 domain for Campaign 1. The **666** term appended to **marketforce** and **gaza** could be another indicator of their relationship, but this could not be confirmed.

Diamond Model

Elastic Security utilizes the Diamond Model to describe high-level relationships between adversaries and victims of intrusions.

Observed Adversary Tactics and Techniques

advanced persistent threats use against enterprise networks.

Tactics

Tactics represent the why of a technique or sub-technique. It is the adversary's tactical goal: the reason for performing an action.

- Resource Development
- Initial Access
- Execution
- Persistence
- Privilege Escalation
- Defense Evasion
- Command and Control

Techniques / Sub techniques

Techniques and Sub techniques represent how an adversary achieves a tactical goal by performing an action.

- Acquire Infrastructure: Domains
- Phishing: Attachment
- Hijack Execution Flow: DLL Side-Loading
- Process Injection
- Scheduled Task
- Native API
- Obfuscated Files or Information: Steganography

Detection

Detection Logic

The following detection rules and behavior prevention events were observed throughout the analysis of this intrusion set.

Behavioral Rules

- NetWire RAT Registry Modification
- Remcos RAT Registry or File Modification

Detection Rules

- Persistence via Scheduled Job Creation
- Command Prompt Network Connection

Signatures

- Windows.Trojan.Parallax
- Windows.Trojan.Netwire
- Windows.Trojan.Remcos

YARA

Elastic Security has created YARA rules to identify this activity.

```
    meta:
        author = “Elastic Security”
        creation_date = "2022-09-05"
        last_modified = "2022-09-15"
        license = “Elastic License v2”
        os = "Windows"
        arch = "x86"
        category_type = "Trojan"
        family = "Parallax"
        threat_name = "Windows.Trojan.Parallax"
    strings:
        $COM_png = { B9 01 00 00 00 6B D1 00 C6 44 15 D4 83 B8 01 00 00 00 C1 E0 00 C6 44 0
        $png_parse = { 8B 4D ?? 8B 04 B8 85 C9 74 ?? 8B F1 90 8A 08 8D 40 ?? 88 0C 1A 42 83
        $config_func = { C7 45 F8 68 74 74 70 8B ?? ?? 8B 02 89 ?? ?? 6A 08 8D ?? ?? 51 E8
        $winnet_function = { B8 77 00 00 00 66 89 ?? ?? B9 69 00 00 00 66 89 ?? ?? BA 6E 00
    condition:
        $config_func or $winnet_function or $COM_png or $png_parse
}

rule Windows_Trojan_Parallax_2 {
    meta:
        author = “Elastic Security”
        creation_date = "2022-09-08"
        last_modified = "2022-09-08"
        license = “Elastic License v2”
        os = "Windows"
        arch = "x86"
        category_type = "Trojan"
        family = "Parallax"
        threat_name = "Windows.Trojan.Parallax"
    strings:
        $parallax_payload_strings_0 = "[Ctrl +" ascii wide fullword
        $parallax_payload_strings_1 = "[Ctrl]" ascii wide fullword
        $parallax_payload_strings_2 = "Clipboard Start" ascii wide fullword
        $parallax_payload_strings_3 = "[Clipboard End]" ascii wide fullword
        $parallax_payload_strings_4 = "UN.vbs" ascii wide fullword
        $parallax_payload_strings_6 = "lt]" ascii wide fullword
        $parallax_payload_strings_7 = ".DeleteFile(Wscript.ScriptFullName)" ascii wide full
        $parallax_payload_strings_8 = ".DeleteFolder" ascii wide fullword
        $parallax_payload_strings_9 = ".DeleteFile " ascii wide fullword
        $parallax_payload_strings_10 = "Scripting.FileSystemObject" ascii wide fullword
        $parallax_payload_strings_11 = "On Error Resume Next" ascii wide fullword
        $parallax_payload_strings_12 = "= CreateObject" ascii wide fullword
        $parallax_payload_strings_13 = ".FileExists" ascii wide fullword
    condition:
        7 of ($parallax_payload_strings_*)
}
```

PARALLAX Payload Extractor

Automating the payload extraction from PARALLAX is a key aspect when it comes to threat hunting as it gives visibility of the campaign and the malware deployed by the threat actors which enable us to discover new unknown samples in a timely manner.

Our extractor takes either a directory of samples with **-d** option or **-f** for a single sample, You can use the **-o** switch to set the output directory of the payloads.

To enable the community to further defend themselves against existing and new variants of the PARALLAX loader, we are making the payload extractor open source under the Apache 2 License. The payload extractor documentation and binary download can be accessed **here**.

Conclusion

In the above research, we have analyzed the two campaigns that we’ve tracked using macro-embedded lure documents that download seemingly benign artifacts from the staging hosts on the Internet, and weaponize

We also highlighted the elements used to cluster the two campaigns together and how the campaigns can be used with analytical models to impose costs on the campaign owners.

References

The following were referenced throughout the above research:

- https://blog.morphisec.com/parallax-rat-active-status
- https://malpedia.caad.fkie.fraunhofer.de/details/win.parallax
- https://attack.mitre.org/software/S0198/
- https://attack.mitre.org/groups/G0120/
- https://malpedia.caad.fkie.fraunhofer.de/actor/evilnum
- http://blog.nsfocus.net/darkcasino-apt-evilnum/

Indicators

Artifacts are also available for **download** in both ECS and STIX format in a combined zip bundle.

Name	STIX 2.1 Indicator Type	Identifier
bc9f19ae835d975de9aaea7d233b6ea9b2bc30f80d192af2e8e68542b588917e	SHA-256	Brian_Tax_Docs.doc lure document

b[illegible]1a0eb2756754ecc548f5b30c2203a5		EXT.doc lure document
9dd709cb989d985a6cfee4a254f894a3b878a03962dbf253cb09a24ece455d58	SHA-256	All Docs.doc lure document
16227f50bbe42a13a2abf0bf0e146f356863de59525c54909ea8ccc2db448f77	SHA-256	msi.dll PARALLAX loader / NETWIRE
0c8c431a1f589fdcf453c7afada63c2e2e2a887e49abdbb222983fa6044fdf66	SHA-256	cs16.wav (shellcode)
6ed65beb692301af5296ba6751063ae40e91c4e69ced43560c67ce58165c36b5	SHA-256	cs16.cfg (config for PNG stage)
5f259757741757c78bfb9dab2cd558aaa8403951c1495dc86735ca73c33d877f	SHA-256	paper.png (stager for NETWIRE)
321d840a23b54bb022ff3a5dcac837e7aec14f66e3ec5e6da5bfeebec927a46c	SHA-256	2021-EXTENSION.doc lure document
443879ee2cb3d572bb928d0831be0771c7120968e442bafe713a6e0f803e8cd9	SHA-256	msvcr100.dll PARALLAX loader / NETWIRE
globalartisticservices[.]com	domain-name	PARALLAX loader domain
DigitalRotPrevention[.]com	domain-name	PARALLAX loader domain
InternationalMusicServices[.]com	domain-name	PARALLAX loader domain
ywiry[.]com	domain-name	NETWIRE C2 domain
ohioohioa[.]com	domain-name	NETWIRE C2 domain
septton[.]com	domain-name	NETWIRE C2 domain

elastic security labs

3.amazonaws[.]com		
mikemikemic[.]com	domain-name	Domains registered by marketforce666@yandex[.]com
ppl-biz[.]com	domain-name	Domains registered by marketforce666@yandex[.]com
opnarchitect[.]net	domain-name	Domains registered by marketforce666@yandex[.]com
micsupportcenter[.]com	domain-name	PARALLAX loader domain
russnet123@protonmail[.]com	email-addr	PARALLAX loader domain registration email address
chisholm.i@aol[.]com	email-addr	NETWIRE C2 domain registration email address

Share this article

Twitter Facebook LinkedIn Reddit