

# Cyble - Xloader Returns with New Infection Technique

By cybleinc

Published: 2022-07-01 · Archived: 2026-04-05 23:06:09 UTC

Cyble analyzes Xloader, a sophisticated malware variant that uses multiple stages for its payload delivery.

## Multistage Delivery of Malware Using Steganography

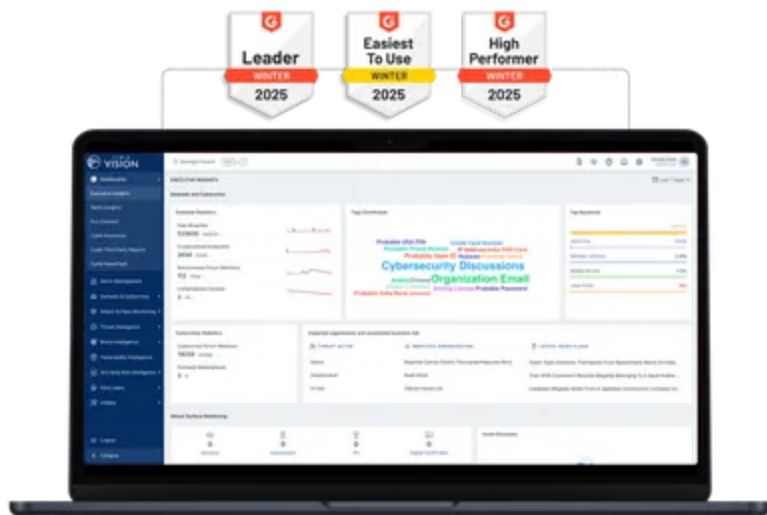
During our routine threat-hunting exercise, Cyble Research Labs came across a [Twitter](#) post wherein a researcher mentioned an interesting infection chain of Xloader malware.

The malware uses multiple file types such as PDF, XLSX, and RTF for its initial infection and execution. It is also designed to drop three modules in memory and execute the final payload using the Process-Hollowing technique. Additionally, The malware uses steganography to hide its malicious content in a bitmap file.

The below figure shows the infection chain of Xloader [malware](#).

### See Cyble in Action

World's Best AI-Native Threat Intelligence



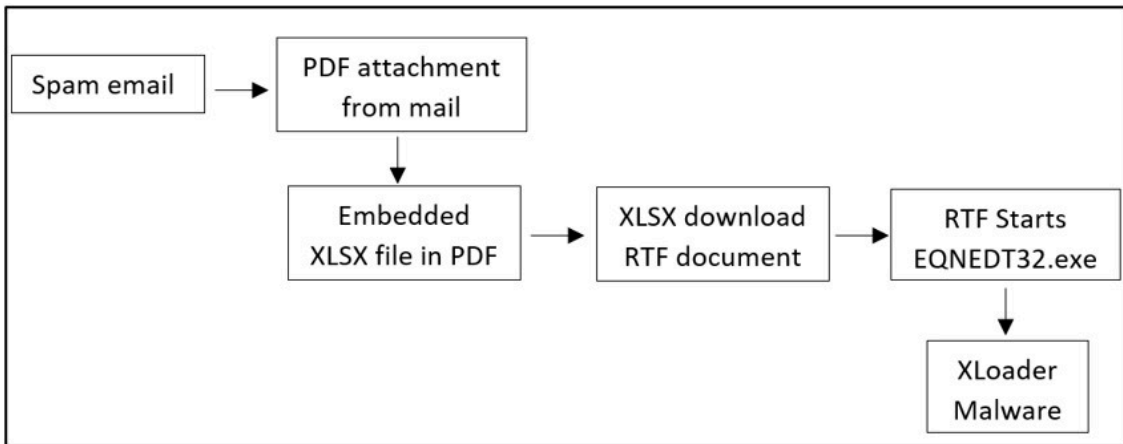


Figure 1 – Xloader Infection Chain

Xloader is a rebranded version of the Formbook stealer. It is designed as a malicious tool to steal [credentials from different web](#) browsers, collect screenshots, monitor and log keystrokes from the victim’s machine, and send them to Command and Control (C&C) server. Typically, Xloader spreads via spam emails that trick victims into downloading a malicious attachment file, such as MS Office documents, PDF documents, etc.

This blog showcases the deep-dive analysis of the malware infection, starting with a spam email containing a PDF attachment to deliver the final payload of Xloader malware. The PDF attachment is shown below.

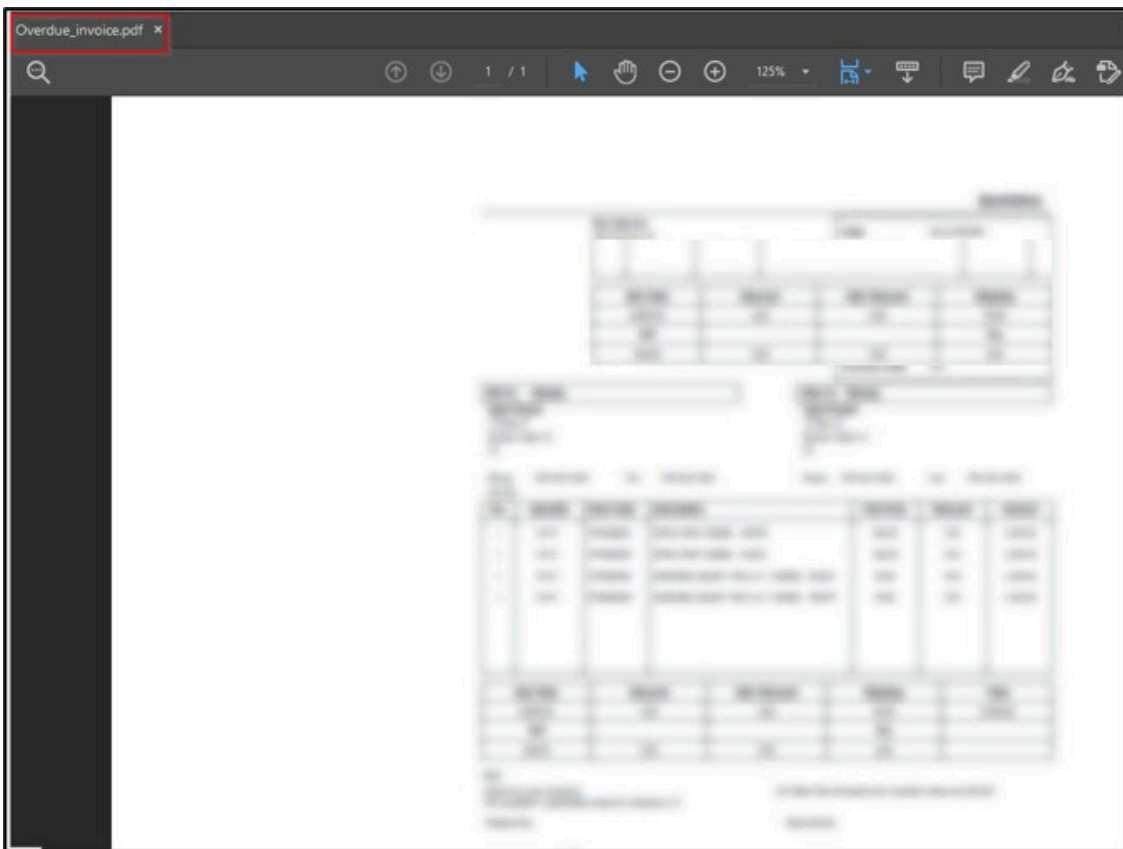
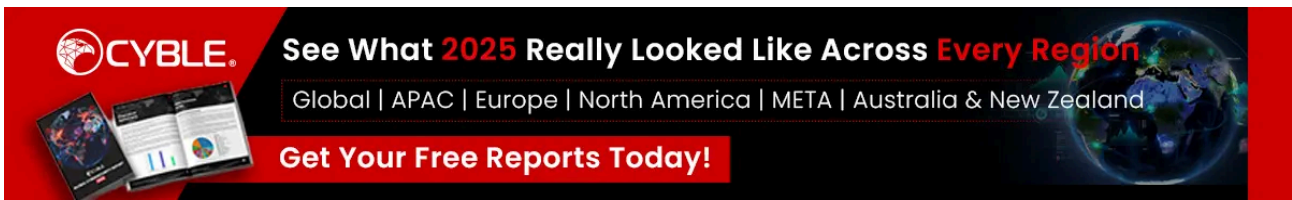


Figure 2 – PDF Attachment from Spam Email

Upon opening a PDF file, it drops the embedded XLSX file named “has been verified. However PDF, JPG, Docx, .xlsx” into the “Temp” location. It then uses multiple extensions of different file formats to trick the user. The

below figure shows the embedded file details of the PDF document.



```
PDFiD 0.2.7 C:\Users\... \Overdue_invoice.pdf
PDF Header: %PDF-1.5
obj                28
endobj             28
stream            26
endstream         26
xref              0
trailer           0
startxref        1
/Page            0
/Encrypt         0
/ObjStm          1
/JS              0
/JavaScript      0
/AA              0
/OpenAction      1
/AcroForm        1
/JBIG2Decode     0
/RichMedia       0
/Launch          0
/EmbeddedFile    1
/XFA             0
/URI             0
/Colors > 2^24   0
```

Figure 3 – Embedded file in PDF Document

Upon execution of the XLSX file, it downloads the RTF document file from the URL – **hxxps[:]//htmlpreview[.]github[.]io@oshi[.]at/Nmtw.**

When the RTF document is opened, MS Word's equation editor (EQNEDT32.exe) will automatically launch and download a .NET malware file from the URL – **hxxp[:]//192.227.173[.]33/71/vbc[.]exe.**

The below figure shows the opened RTF document.

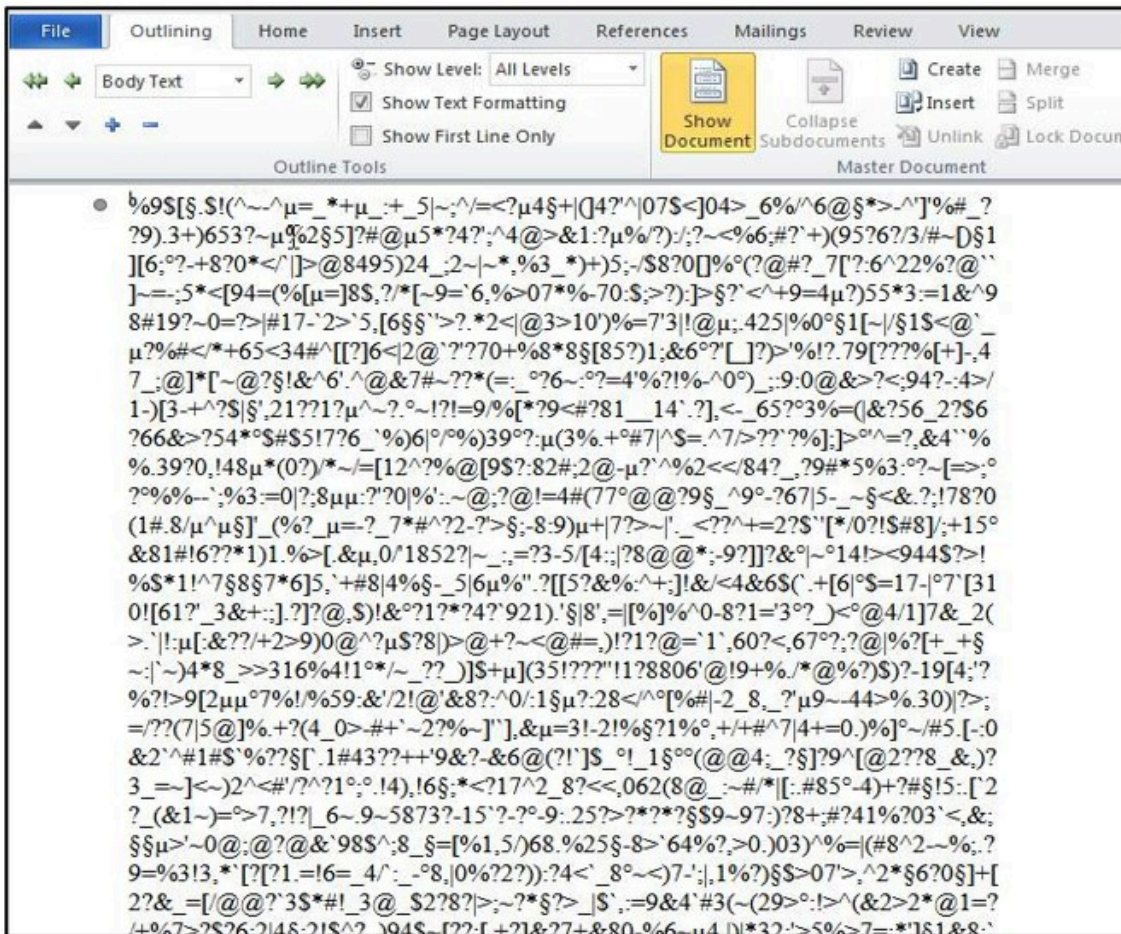


Figure 4 – RTF Document

The .NET executable file named “vbc.exe” is downloaded from the RTF document via equation editor vulnerability (CVE-2017-11882) and is an obfuscated binary file. The below figure shows the obfuscated and de-obfuscated file details such as methods and functions.

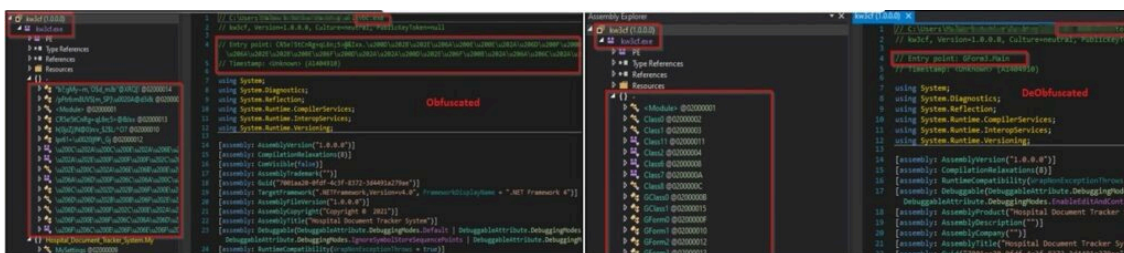


Figure 5 – Obfuscated and De-obfuscated details of the “vbc.exe” file

## Technical analysis:

We have taken the sample hash (SHA256), **d0c85ba5e6d88e1e0b5f068f125829b4e224b90be2488f2c21317447dc51fb9e** for our analysis. It is a 32-bit, .NET executable file named as “vbc.exe”.

Upon execution of the vbc.exe file, the method *Convert.FromBase64String()* in the *Main()* function decodes the base64 string content and returns a new PE file, as shown below.

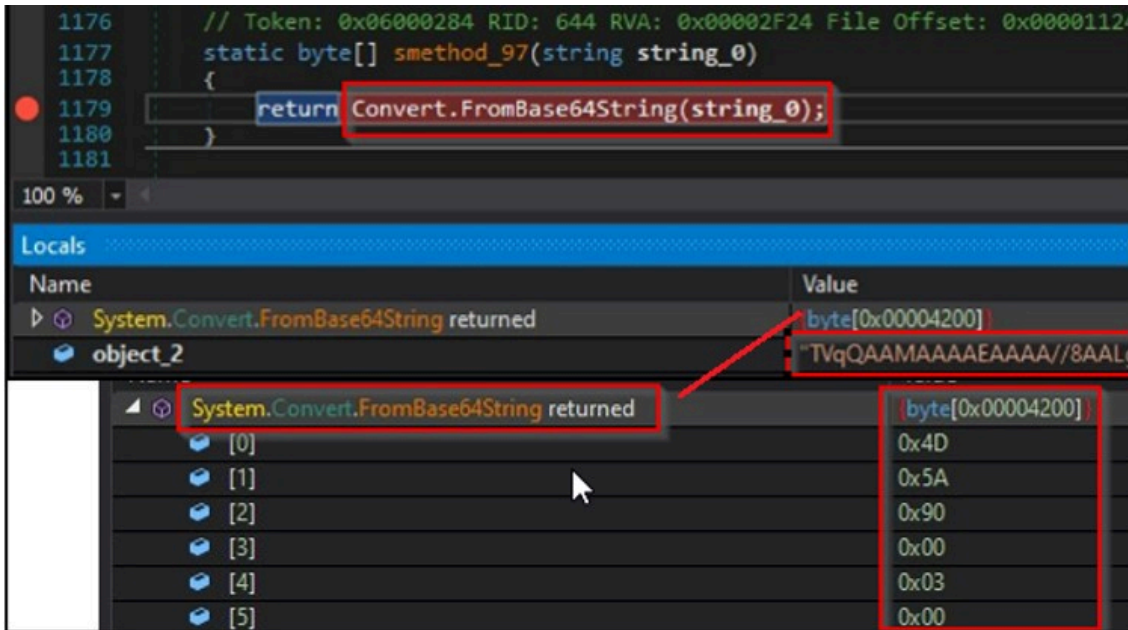


Figure 6 – Base64 String Conversion

After decoding the base64 content, vbc.exe loads the converted PE module named “*Bunifu.UI.dll*” into memory by using a dynamically invoked function with passing arguments of strings such as “*Invoke*” and “*Bunifu\_TextBox*.” The below figure shows the concatenated strings used in the malware file.

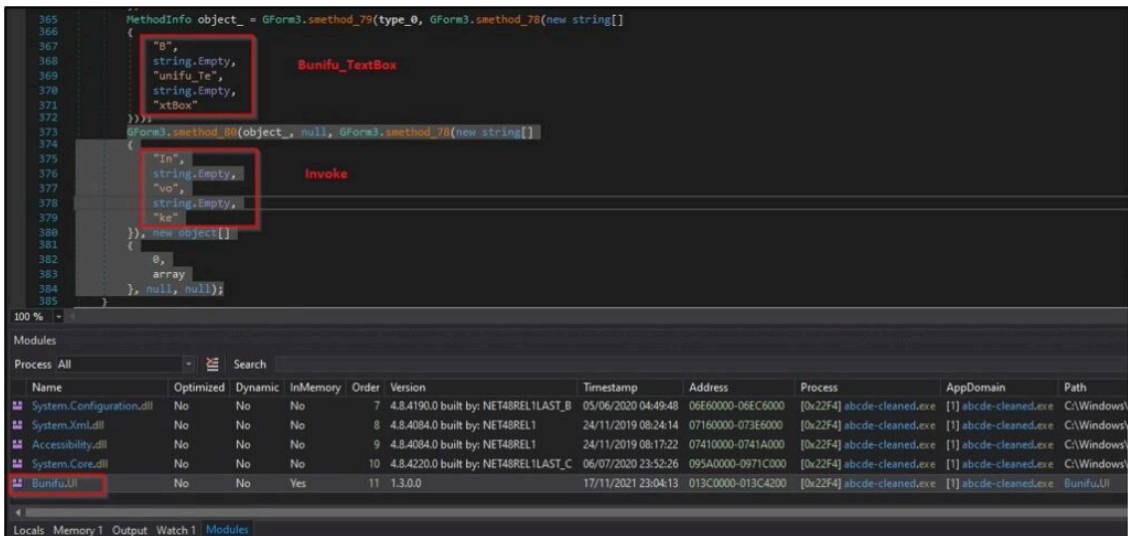


Figure 7 – String Concatenation

The module “*Bunifu.UI.dll*” is also an obfuscated .NET file. The below figure shows the de-obfuscated content of the new assembly file and runs the *Bunifu\_TextBox()* function, which retrieves the embedded bitmap image “*QQvruB*” present in the resource (“*Hospital\_Document\_Tracker\_System.Resources.resources*”) of the parent malware vbc.exe file. It then calls the *Sleep* function to delay the execution before accessing the resource for the bitmap image.

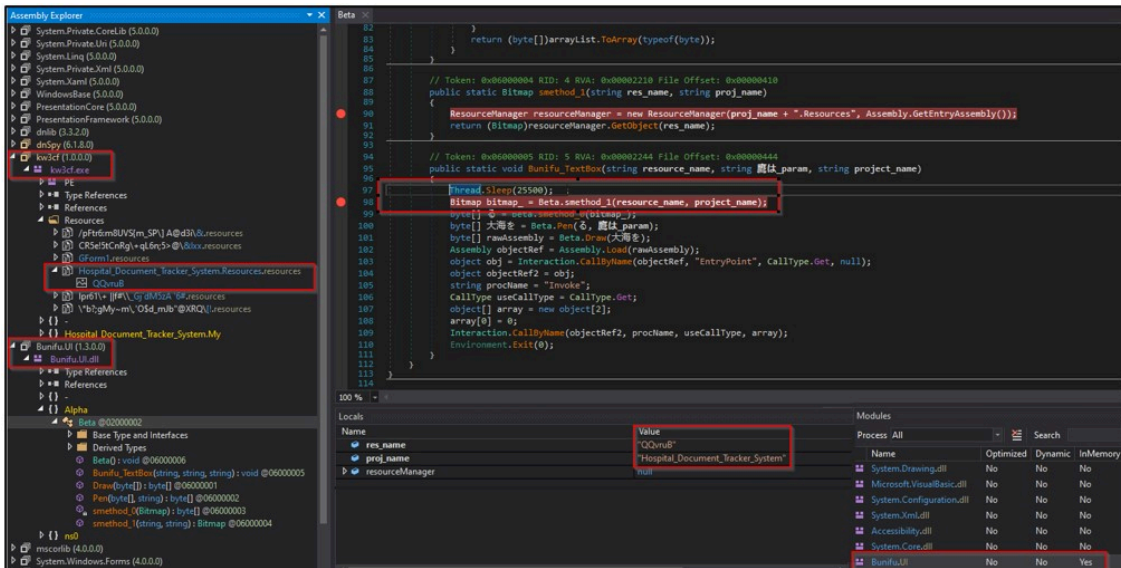


Figure 8 – De-obfuscated Content of New Module “Bunifu.UI.dll”

The malware uses the steganography technique to hide malicious content in the compressed bitmap image embedded in the resource of the parent malware file vbc.exe, shown below.

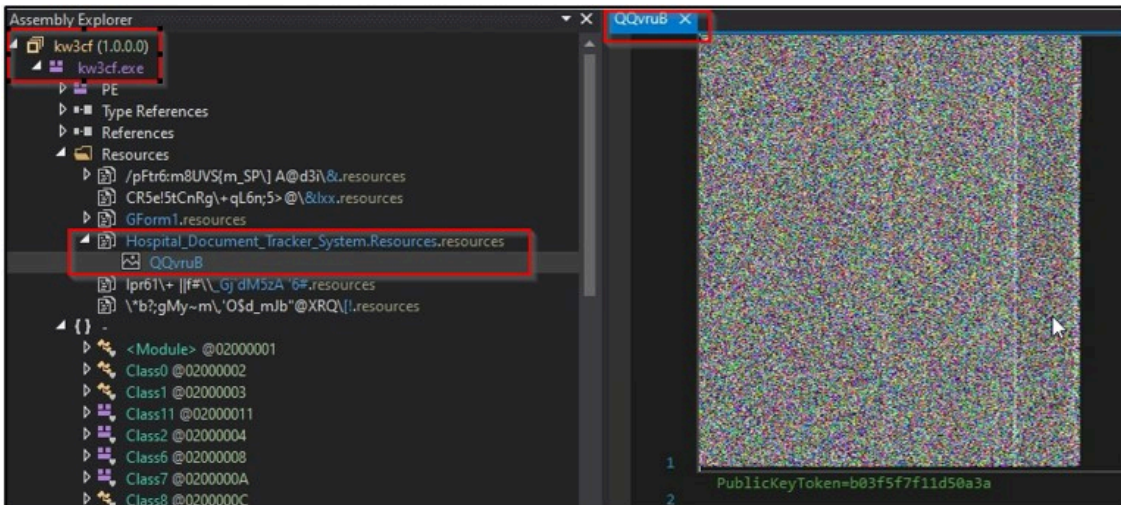


Figure 9 – Compressed Bitmap Embedded in Main File Resource

The successful decompression of the bitmap image retrieves another .NET file in memory, as shown in Figure 10. The “Bunifu.UI.dll” module loads the new binary using the *Assembly.Load* method by passing the decompressed bitmap content as an argument.

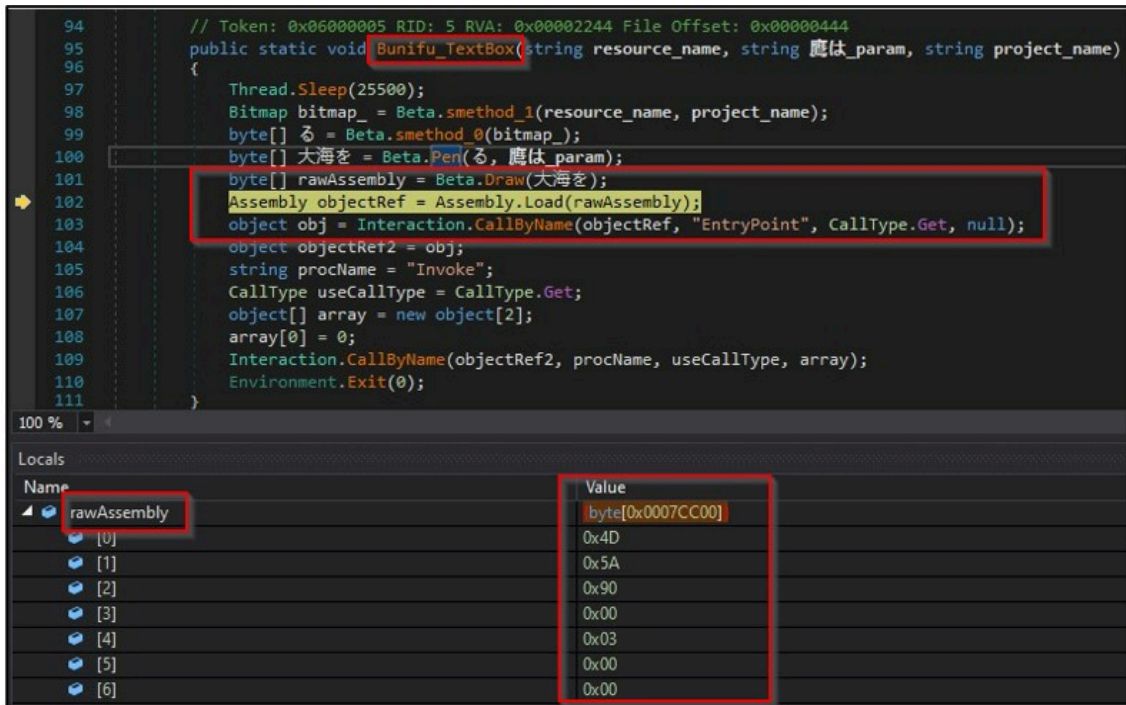


Figure 10 – Decompressed Bitmap Content of New Module from Resource

The main purpose of “*Bunifu.UI.dll*” is to decompress the bitmap image from a resource using the “*GZipStream*” class, as shown in the figure below.

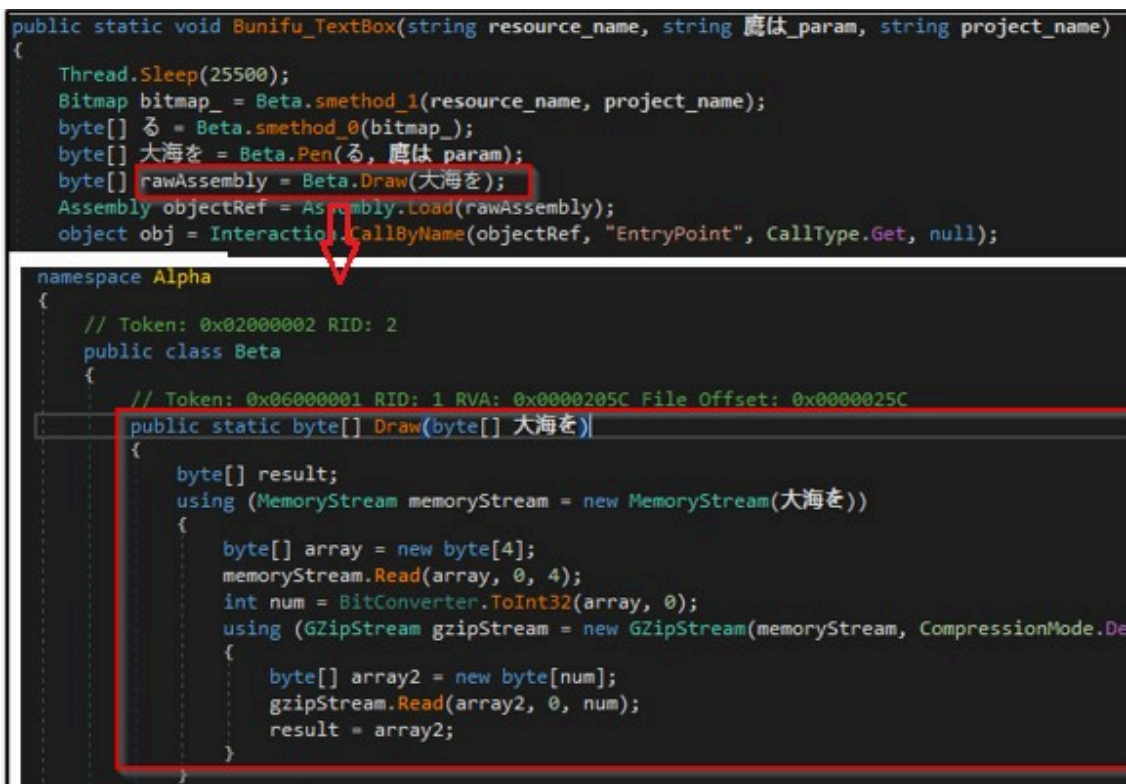


Figure 11 – Decompression Function

The new file decompressed from the resource is another obfuscated .NET binary titled “*MajorRevision.exe.*” The figure below shows the newly loaded module in memory with the module name in the Chinese script.

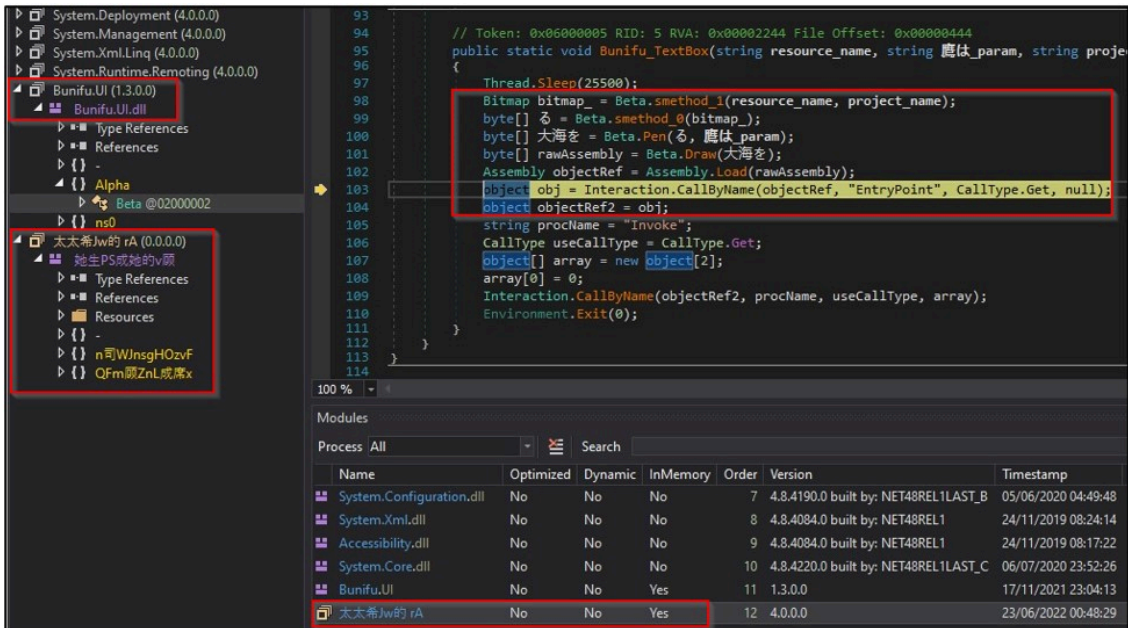


Figure 12 – Loaded New Module “MajorRevision.exe”

The below figure shows the de-obfuscated “MajorRevision.exe” assembly file.

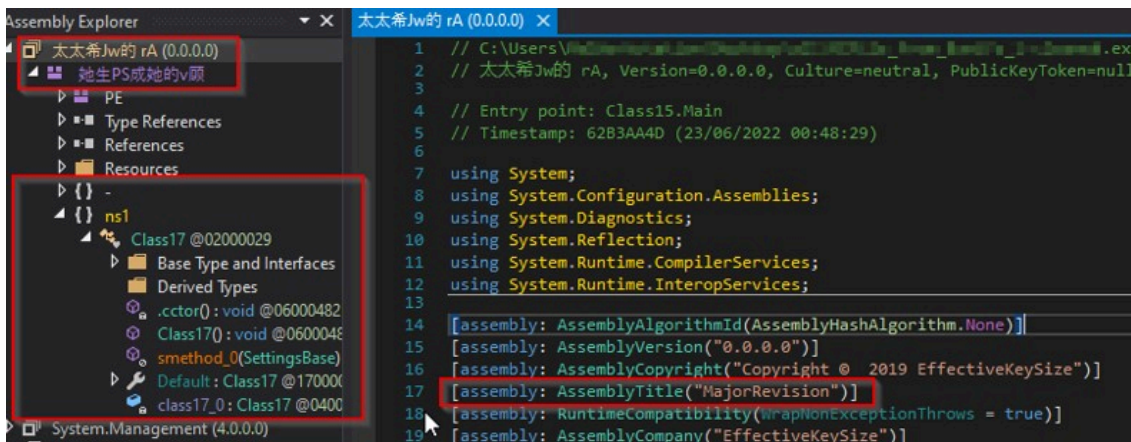


Figure 13 – De-obfuscated MajorRevision.exe File

Upon execution of the “MajorRevision.exe” module, it first creates a mutex named “fBEQVtAy” to ensure that only one instance of malware runs on the victims’ system. The malware exits if the mutex is already present.

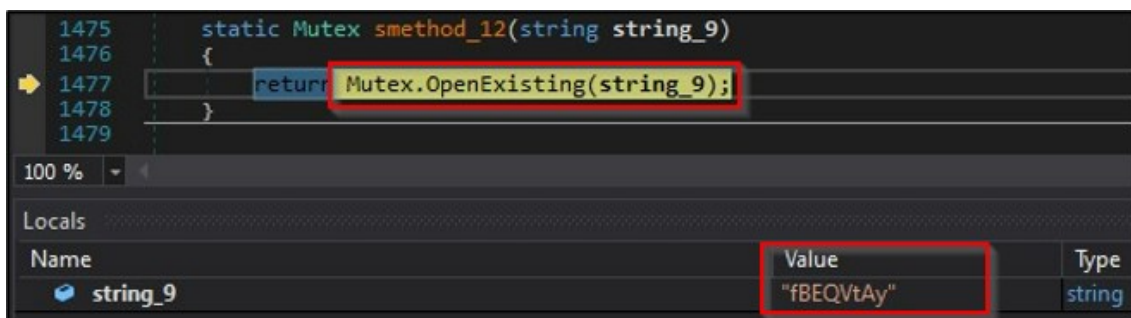


Figure 14 – Mutex Creation



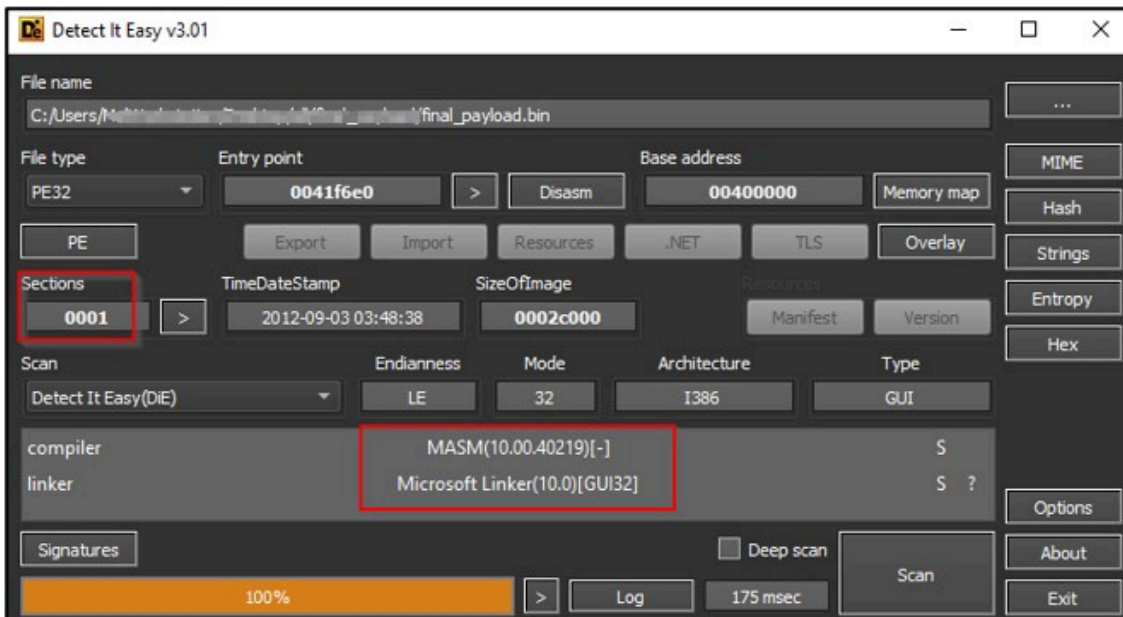


Figure 17 – Final Payload Details

Xloader malware uses the magic bytes “XLNG,” shown in the figure below.

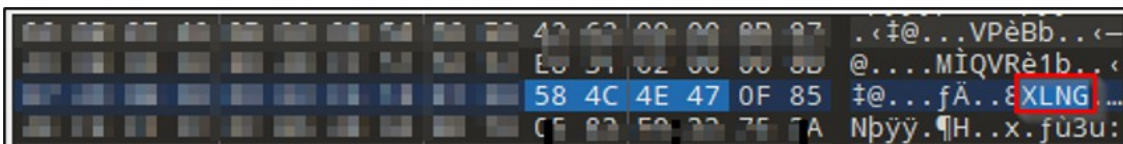


Figure 18 – XLNG Magic Bytes of Xloader

Upon successful execution, Xloader drops an executable file in the following location and injects it into explorer.exe.

- “C:\Program Files (x86)\L9rq\winmrhl7bm.exe”

To establish persistence, the malware creates the below registry key for autorun to execute the dropped malware file when the user logs in to the system every time.

- *HKEY\_LOCAL\_MACHINE*  
 \SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run\J8TPYFN8OVE = “C:\Program Files (x86)\L9rq\winmrhl7bm.exe”

Finally, after a successful connection to the [Threat Actor’s](#) C&C server, Xloader can be instructed to download and launch additional payloads, terminate and uninstall the malware, etc.

Additionally, Xloader steals user credentials or cookies from browsers, logs keystrokes, steals clipboard content, takes screenshots, and sends them to the TA’s C&C server.

## Conclusion

Information stealers are evolving as increasingly sophisticated threats in the cybercrime ecosystem. They can cause severe damage to individuals and organizations in the case of privacy violations, confidential information

leakage, etc.

Exploiting the human element is often easier for [Threat Actors](#) compared to exploiting complex vulnerabilities. Throughout our analysis, we have observed that Xloader looks like a prominent malware variant that is constantly updated by improving its code which adds new features, more obfuscation, the use of anti-analysis techniques, etc.

Cyble Research Labs will closely monitor Xloader malware and other information stealers and analyze them to understand their TTPs better and update our readers accordingly.

## Our Recommendations

- Avoid downloading pirated software from unverified sites.
- Use strong passwords and enforce [multi-factor authentication](#) wherever possible.
- Keep updating your passwords after certain intervals.
- Use a reputed anti-virus and [internet security](#) software package on your connected devices, including PC, laptop, and mobile.
- Refrain from opening untrusted links and email attachments without first verifying their authenticity.
- Block URLs that could be used to spread the malware, e.g., Torrent/Warez.
- Monitor the beacon on the network level to block data exfiltration by malware or TAs.
- Enable Data Loss Prevention (DLP) Solutions on employees' systems.

## MITRE ATT&CK® Techniques

Tactic	Technique ID	Technique Name
Initial Access	<a href="#">T1566</a>	Phishing
Execution	<a href="#">T1204</a> <a href="#">T1203</a>	User Execution Exploitation for Client Execution
Persistence	<a href="#">T1547</a>	Registry Run Keys / Startup Folder
Defence Evasion	<a href="#">T1497</a>	Virtualization/Sandbox Evasion
Credential Access	<a href="#">T1552</a>	Credentials In Files
Lateral Movement	<a href="#">T1021</a>	Remote Services
CNC	<a href="#">T1071</a>	Application Layer Protocol

## Indicator Of Compromise (IOCs)

Indicators	Indicator Type	Description
afa05a84f53f793fdad59d8af603b497	MD5	Spam email
bdbc99cb9698f3754dea53bb192e650b2f0c203c	SHA1	

9d3c9168bc5d52c0372f31565bf2ec690a39cfd52bc76d0ef01083e419da805b	Sha256	
96d95ee6d0c9da16d245579ad1ff2e9f f852ac58b11e6b314271e2afdd33da84fc3cb8d8 6d45a03b32c4a9bab48c75bec8443b5af40ae43e055db77796a6328cb6e87ffe	MD5 SHA1 Sha256	PDF
2fc6db5b63ba91752b946d76b803a4a9 45982471aca75de846442d16c84c5b61caa6c045 30d5632ef75e81aa6a48eae64f2155acc39e64f6367a5c6152e8ec74b44ac6de	MD5 SHA1 Sha256	XLSX
e5cde34f443cab2ebecf850518d0aeeb 375ecc13e71755cc4ab260f518207892e87c55e3 d106de4854f334b826f7ed6e97b02eff34e8ab8ea956d461d67c4225792185a1	MD5 SHA1 Sha256	RTF
1f65d7826fbcc2d6c50f6c493c901588 4290f6b300595e807e8cacd5ff172b0a0f37c845 d0c85ba5e6d88e1e0b5f068f125829b4e224b90be2488f2c21317447dc51fb9e	MD5 SHA1 Sha256	Obfuscated .NET exe Main file
a0dc449956fd7eefaeb204d66b668330 76b958e128a7f2dd052634d5e7dfbf2f67f20ae9 50204673d080635b23b8f219a70e276acd3dd3779543fbd4b82a217c06dc14fb	MD5 SHA1 Sha256	De- obfuscated .NET exe Main file
39f524c1ab0eb76dfd79b2852e5e8c39 428018e1701006744e34480b0029982a76d8a57d 79823e47436e129def4fba8ee225347a05b7bb27477fb1cc8be6dc9e9ce75696	MD5 SHA1 Sha256	Obfuscated .NET exe Stage 1
bc31d889dd60360d38796521b452d775 7e52c29418bd13c749da76506251ad3ad291d06c 32abba85bb16f812822c789882e37cd37c62e15ea0aceade45ead1d93ff012a	MD5 SHA1 Sha256	De- obfuscated .NET exe Stage 1
73aac8ac5dc4ded42398f9fe2a191c19 4f3ed7fa592f4ae4c4462928543dcbd4997f2549 6672b26a03db7ec5d61e90ce7827c422cb6a8a942cc1c77f92f97e263a35d8e5	MD5 SHA1 Sha256	Obfuscated .NET exe Stage 2
0227a4419e2948a886a2e324180f23e6 43c1ee78411b939e19688ff9ea9ebc433d9051a1 c7b2597253067c1169aeef5e04948575bf7df65e1787098cc9afc2e10685acdf	MD5 SHA1 Sha256	De- obfuscated .NET exe Stage 2
7d4539bd445cf9821fd2e05dc0b1107e 964e56a5e1f32101f04fa3fc62ec17c66b3c174e 3b65b859612be75eb528caf7b0cc66bc049fdb062b6b6aa29ea9c356114a4fe	MD5 SHA1 Sha256	Final payload MASM exe
hxxps[:]//htmlpreview[.]github[.]io@oshi[.]at/Nmtw	URL	download RTF

		file from C&C
<code>hxxp[:]//192[.]227[.]173[.]33/71/vbc[.]exe</code>	URL	Download EXE file from C&C

---

Source: <https://blog.cyble.com/2022/07/01/xloader-returns-with-new-infection-technique/>