

Forum: [Hack Forums](#)

Thread: [Squad Mailer Unlocked Clean Version](#)

mr0x03 - April 20, 2025, 8:00 pm

Deliver your emails directly to the inbox – clean, fast, and effective.

Looking for a reliable inbox mailer to power your email marketing or outreach campaigns? Squad Mailer is the latest, most efficient solution out there. It's lightweight, clean (no junk scripts), and optimized for high deliverability.

Whether you're sending newsletters, business offers, or custom campaigns – Squad Mailer makes sure your emails land straight in the inbox, not in spam.

- Clean, easy-to-use interface
- High inbox rate
- Supports HTML & plain text email formats
- Works with major SMTP providers

? Download Squad Mailer: [deleted]

Reviewed by MRT-X : Loader

Bastille - April 20, 2025, 9:01 pm

OP shared what seems to be a mailing tool. Uploading it on VT doesn't trigger many AV : [VirusTotal Report](#)

When uploading on VT, it's important to know that even if some AV are triggered, it doesn't necessarily mean it's malicious :

- some code portion could be related to malicious activities, but the purpose of the program is not malicious
- you need to check WHICH AV have been triggered, the most known / more effective ones can be more reliable (ESET, Kaspersky, Microsoft, ...)
- if there are triggers but from not very famous AV, then it's more likely false positive

Even with a low detection rate I wanted to take a look at it, just to be sure.

I've uploaded it on TRIA.GE aaaaaaand, nothing happened. So I suspected a protection against virtual environment :

[TRIA.GE Report](#)

So I loaded the program on Detect It Easy to see what language has been used, and if it's packed or obfuscated, C and not protected :

As it's not protected I wanted to take a look through disassembling, so I launched Ghidra. Browsing the defined strings with usual keywords I've found the following :

I've noticed that the usual stuff for anti-analysis are also present :

Code IsDebuggerPresent

CheckRemoteDebuggerPresent

GetTickCount

QueryPerformanceCounter

Defanged URL : `hxxps[:]informsattelecom[.]com[.]br/loader2234[.]png` / Not sure about the website tho, if it has been hacked to host the payload or not. Using a 3 month cert

So I've downloaded the payload with the full URL (VPN enabled), it took few minutes, maybe it's a protection, or because of the VPN :

Code `wget hxxps[:]informsattelecom[.]com[.]br/loader2234[.]png`

Here is the VirusTotal Report :

<https://www.virustotal.com/gui/file/ba1567468ac271a9ddd5814b0449ea7ced006e0d5df5745cf186e8227355ca97?nocache=1> (more suspicious), with one label that comes very often :

Generic.ShellCode.Donut.Marte

First time I come across this kind of thing, I could take a deeper look but don't want to spend too much time, if someone have more info about "Donuts" and how to analyze such files I'm interested ?

DarkG8 - April 22, 2025, 8:04 am

(Apr 20, 2025 09:01 PM) Bastille Wrote: [\[CITE\]](#) OP shared what seems to be a mailing tool.

Uploading it on VT doesn't trigger many AV : [VirusTotal Report](#)

When uploading on VT, it's important to know that even if some AV are triggered, it doesn't necessarily mean it's malicious :

- some code portion could be related to malicious activities, but the purpose of the program is not malicious
- you need to check WHICH AV have been triggered, the most known / more effective ones can be more reliable (ESET, Kaspersky, Microsoft, ...)
- if there are triggers but from not very famous AV, then it's more likely false positive

Even with a low detection rate I wanted to take a look at it, just to be sure.

I've uploaded it on TRIA.GE aaaaaaand, nothing happened. So I suspected a protection against virtual environment :

[TRIA.GE Report](#)

So I loaded the program on Detect It Easy to see what language has been used, and if it's packed or obfuscated, C and not protected :

As it's not protected I wanted to take a look through disassembling, so I launched Ghidra. Browsing the defined strings with usual keywords I've found the following :

I've noticed that the usual stuff for anti-analysis are also present :

Code IsDebuggerPresent

CheckRemoteDebuggerPresent

GetTickCount

QueryPerformanceCounter

Defanged URL : `hxxps[:]informsatelecom[.]com[.]br/loader2234[.]png` / Not sure about the website tho, if it has been hacked to host the payload or not. Using a 3 month cert

So I've downloaded the payload with the full URL (VPN enabled), it took few minutes, maybe it's a protection, or because of the VPN :

Code `wget hxxps[:]informsatelecom[.]com[.]br/loader2234[.]png`

Here is the VirusTotal Report :

<https://www.virustotal.com/gui/file/ba1567468ac271a9ddd5814b0449ea7ced006e0d5df5745cf186e8227355ca97?nocache=1> (more suspicious), with one label that comes very often :

Generic.ShellCode.Donut.Marte

First time I come across this kind of thing, I could take a deeper look but don't want to spend too much time, if someone have more info about "Donuts" and how to analyze such files I'm interested ? You've really piqued my interest and I decided to look into this in detail:

Donut is a tool for generating position-independent shellcode from .NET assemblies (EXE/DLL) that:

Loads and executes code in memory (without writing to disk).

Bypasses AMSI, ETW, AppLocker and other security mechanisms.

Supports encryption and anti-debugging.

Static analysis:.

1]File type detection:

Code file Generic.ShellCode.Donut.Marte # Check the signature

strings Generic.ShellCode.Donut.Marte | less # Searching for string artifacts

If the output contains Donut or MZ (PE header), it is a shell code.

2. Structure parsing (if it is PE):

PEiD / PESTudio - search for packers (UPX, Donut).

Detect It Easy (DIE) - signature analysis.

3. Decompile .NET (if the source is .NET):

Code dnSpy Generic.ShellCode.Donut.Marte # For .NET builds

Dynamic analysis:

1. Sandbox Run:Any.Run - Automatic Behavior Analysis.

Hybrid Analysis

2. manual debugging:x64dbg / OllyDbg - to analyze shell code.

Process Hacker - injection monitoring.

3. Donut emulation:

If the file is pure Donut shellcode, you can use:

Code donut -f Generic.ShellCode.Donut.Marte -o decoded.bin # Decryption (if not encrypted)
Tools for analyzing Donut:

Tool - For what: Donut (GitHub) - Shell code generation/decoding
PE-sieve - Search for in-memory injections
Capa (FireEye) - Analyze shell code features
UnpacMe - Unpack packaged files

What's in the end:
Such files are shellcode, often used for stealthy code execution.

What I recommend: Static analysis: strings, PEiD, dnSpy.
Dynamic analysis: Any.Run, x64dbg.
Donut decompilation: Code donut -f <file> -o decoded.bin
That's all I've learned on the internet and from my knowledge.

ginxforce - May 5, 2025, 2:47 am
