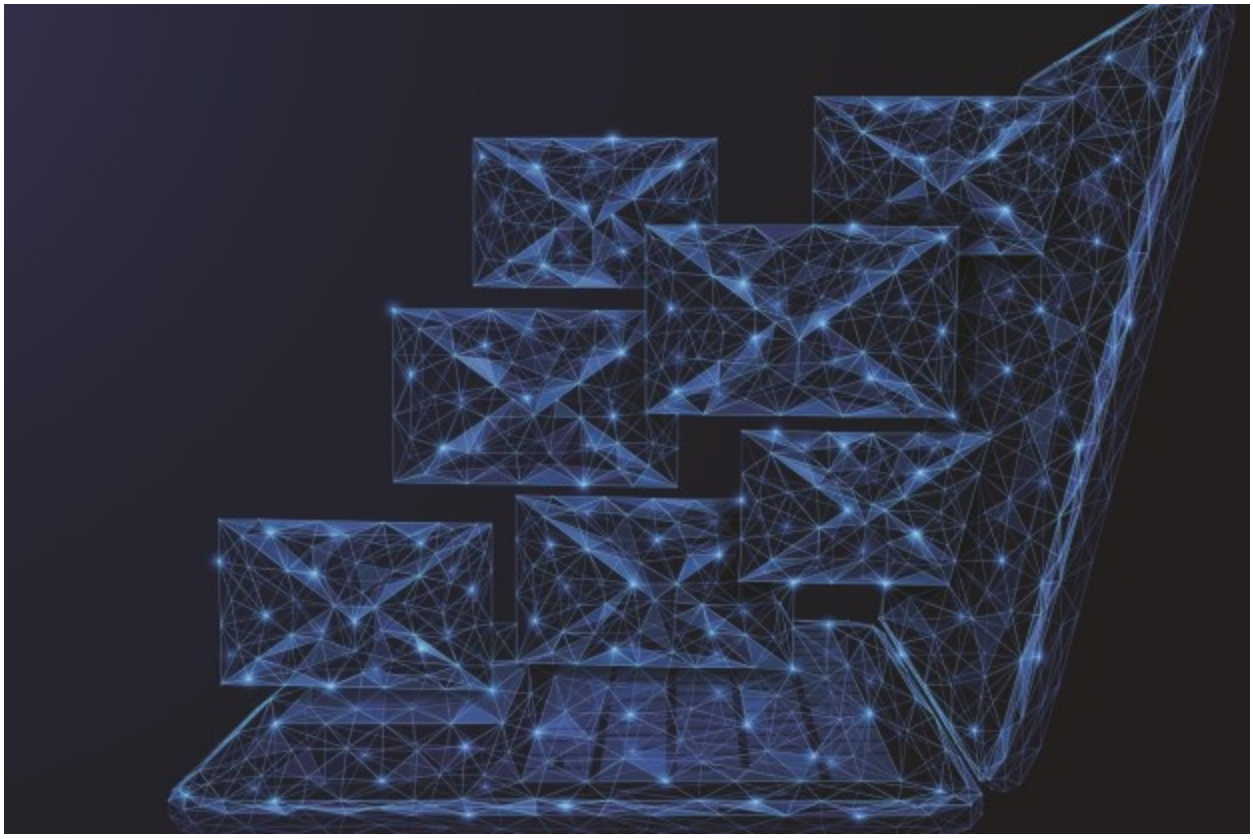
How Emotet is changing tactics in response to Microsoft's tightening of Office macro security

welivesecurity.com/2022/06/16/how-emotet-is-changing-tactics-microsoft-tightening-office-macro-security/

June 16, 2022



Emotet malware is back with ferocious vigor, according to ESET telemetry in the first four months of 2022. Will it survive the ever-tightening controls on macro-enabled documents?

Rene Holt

16 Jun 2022 - 11:30AM

Emotet malware is back with ferocious vigor, according to ESET telemetry in the first four months of 2022. Will it survive the ever-tightening controls on macro-enabled documents?

One of the key findings from the ESET Threat Report T1 2022 is that the Emotet botnet has risen, Phoenix-like, from the ashes, pumping out vast amounts of spam in March and April 2022, to the point that its detections grew more than a hundredfold in the first four months of 2022 compared to the last four months of 2021. Much of this activity involved Word documents tainted with malicious macros.

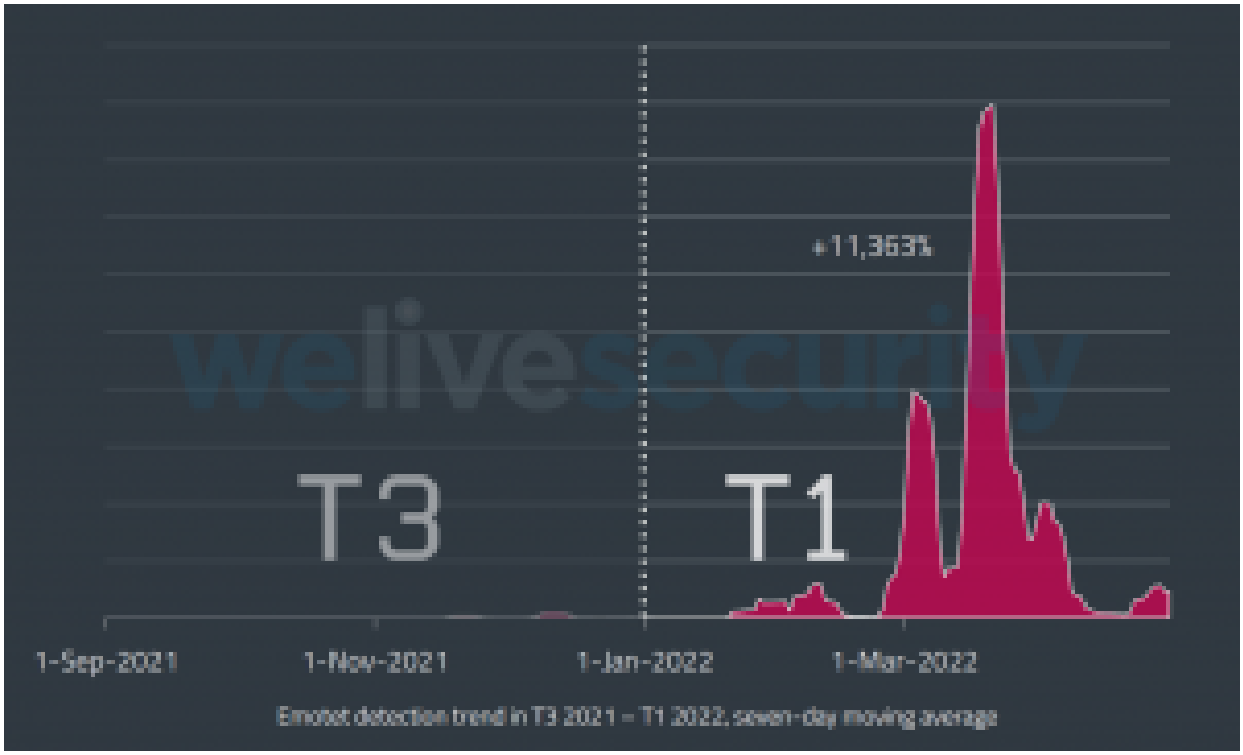


Figure 1. Emotet detections in ESET telemetry

Because Microsoft is tightening up the default handling of macro-enabled files, finagling recipients into clicking "Enable Content" will not remain a viable tactic for long. What does this mean for Emotet? Could this extremely pervasive threat even sink into oblivion barely a few months after it shook off the effects of the law enforcement operation hailed as one of the largest of its kind ever?

Not so fast – Emotet's operators aren't known for resting on their laurels.

Emotet – a macro view

First sighted as a banking trojan in June 2014, Emotet has since changed drastically into a crime-as-a-service platform, selling access to compromised systems to other criminal groups. Thus, once Emotet is running on a computer, it typically downloads and executes other strains of malware, such as Dridex, Gootkit, IcedId, Nymaim, Qbot, TrickBot, Ursnif, and Zbot.

Emotet has a modular program design, with a main module that is disseminated through vast spam campaigns that distribute emails containing malicious Microsoft Word documents. Emotet then uses additional modules to:

- *spread further by assembling and delivering spam emails*
- *spread to nearby, insecure Wi-Fi networks by compromising connected users*
- *brute-force network share usernames and passwords*
- *turn compromised systems into proxies within its command-and-control infrastructure*
- *abuse legitimate Nirsoft applications, such as MailPassView and WebBrowserView, that can recover passwords from popular email clients and web browsers, respectively.*
- *steal email addresses and names from the compromised system's Microsoft Outlook instance*
- *steal all email messages and attachments from compromised systems*

In 2018, Emotet resuscitated an effective technique – email thread hijacking – to increase the likelihood of a potential victim opening the email attachments. It started stealing email conversations found in compromised systems' inboxes and reusing them in its spam campaigns. This is, of course, a very effective way of adding legitimacy to a malicious email:

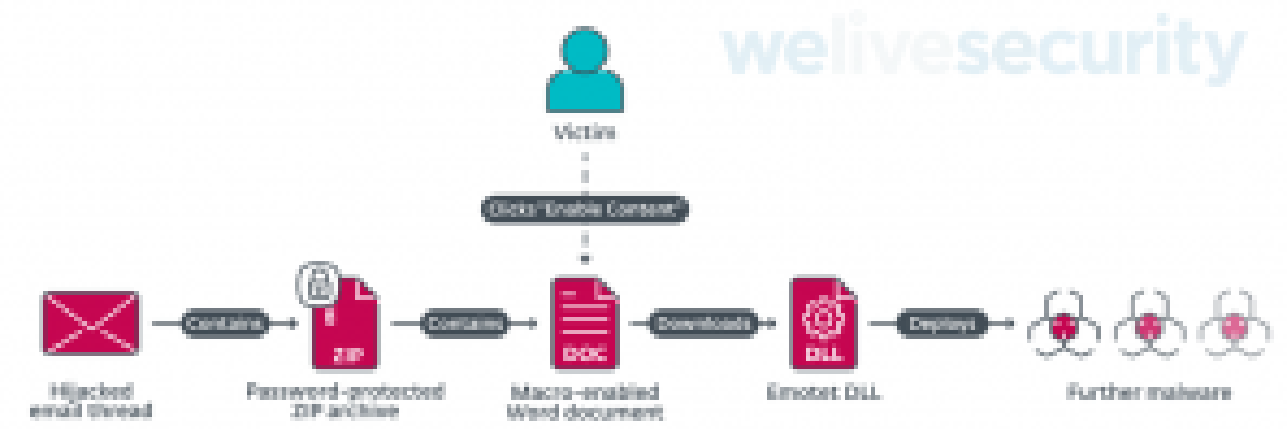


Figure 2. Emotet's operators use macro-enabled Word documents to deliver malware

Should the victim extract the macro-laden Word document from the ZIP archive, open it, and then click "Enable Content", the malicious macros can run, ultimately downloading Emotet.

Microsoft's move (on February 30th 2022, so to speak) to throw out the "Enable Content" button came at a time for Emotet when, after recovering from last year's takedown efforts, it had been churning out spam campaigns en masse in March and April 2022. Taking note of the change, Emotet's developers have shifted to experimenting with different techniques to replace their dependence on macros as the initial code stage of their malware delivery platform.

Emotet shifting techniques

Between April 26th and May 2nd, 2022, ESET researchers picked up a test campaign run by Emotet operators where they replaced the typical Microsoft Word document with a shortcut (LNK) file as the malicious attachment.

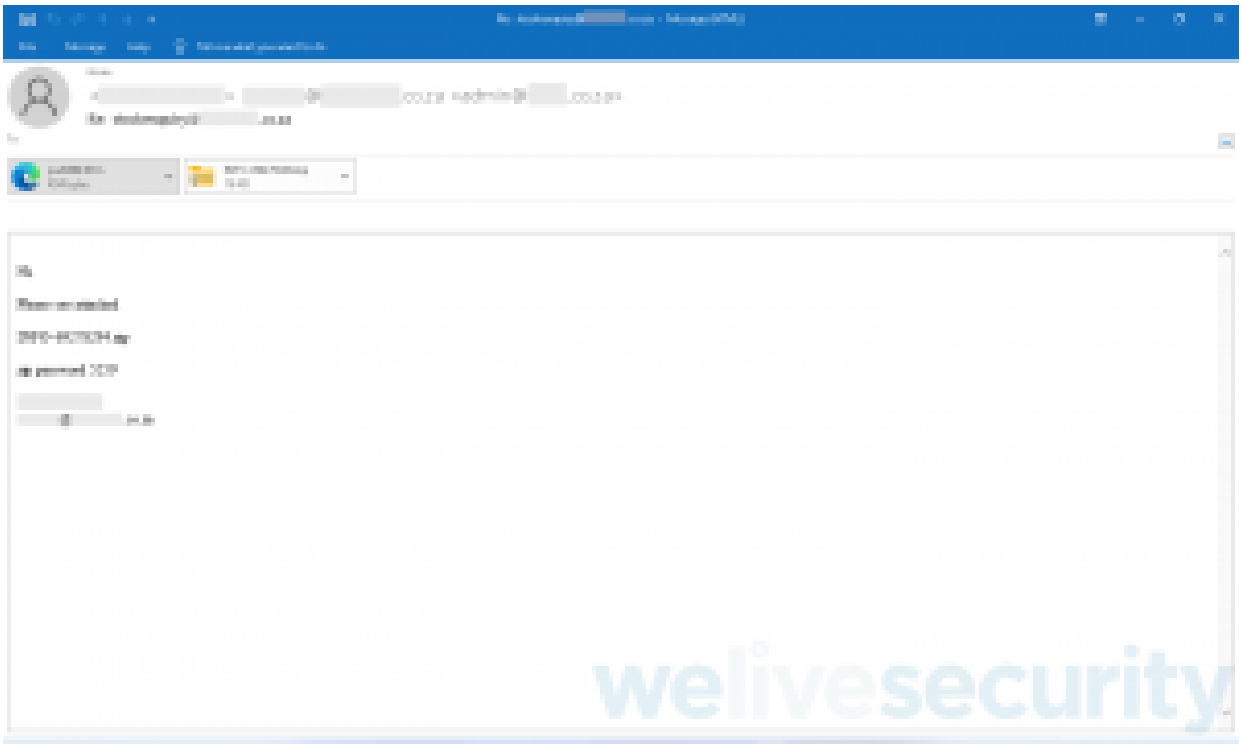


Figure 3. A malicious email sent by Emotet's operators

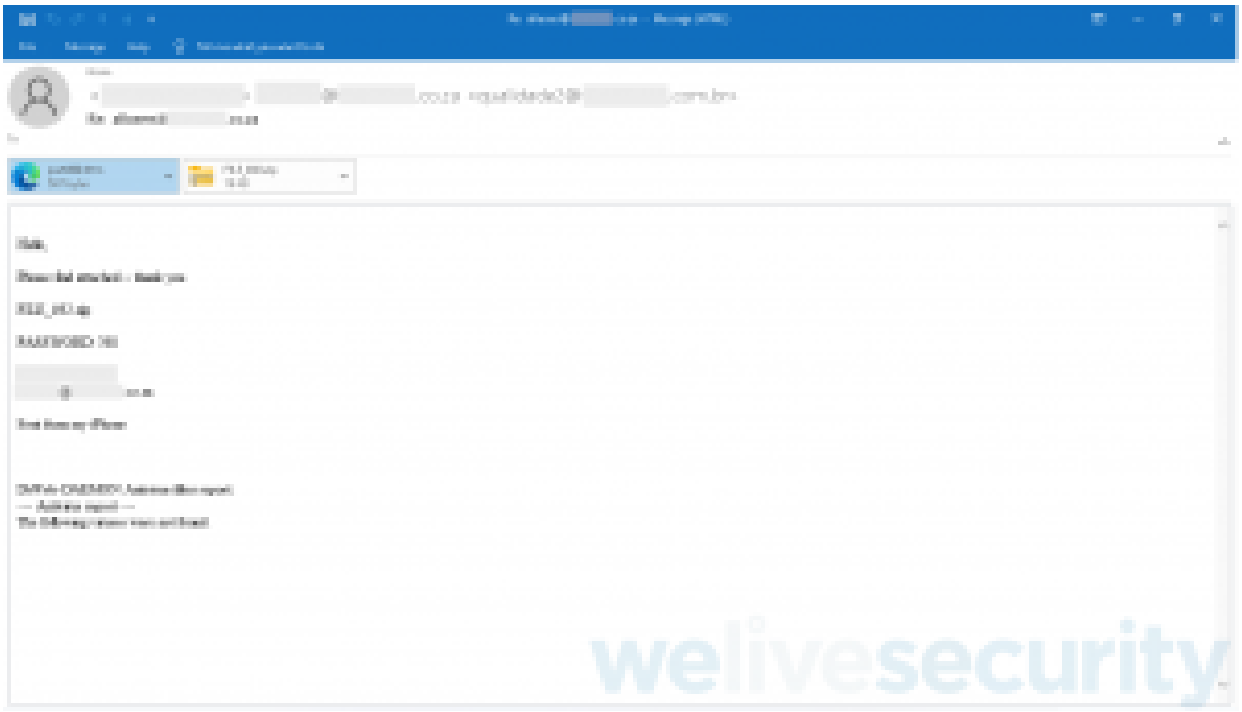


Figure 4. Another malicious email sent by Emotet's operators

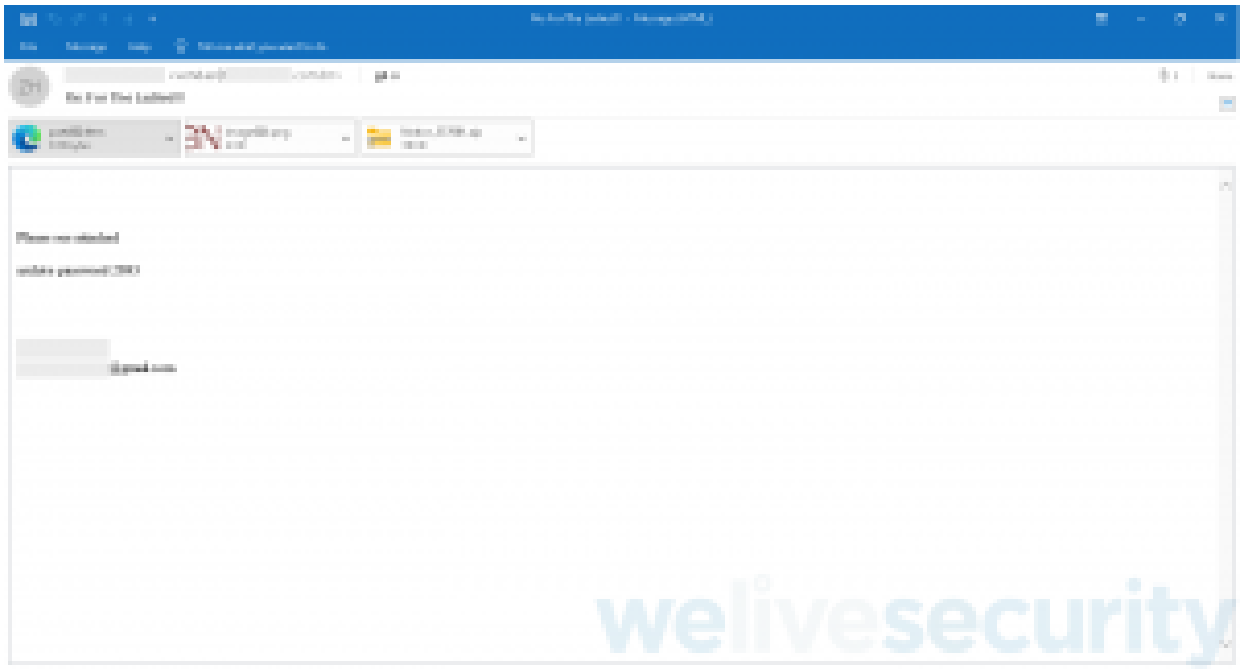


Figure 5. Yet another malicious email sent by Emotet's operators

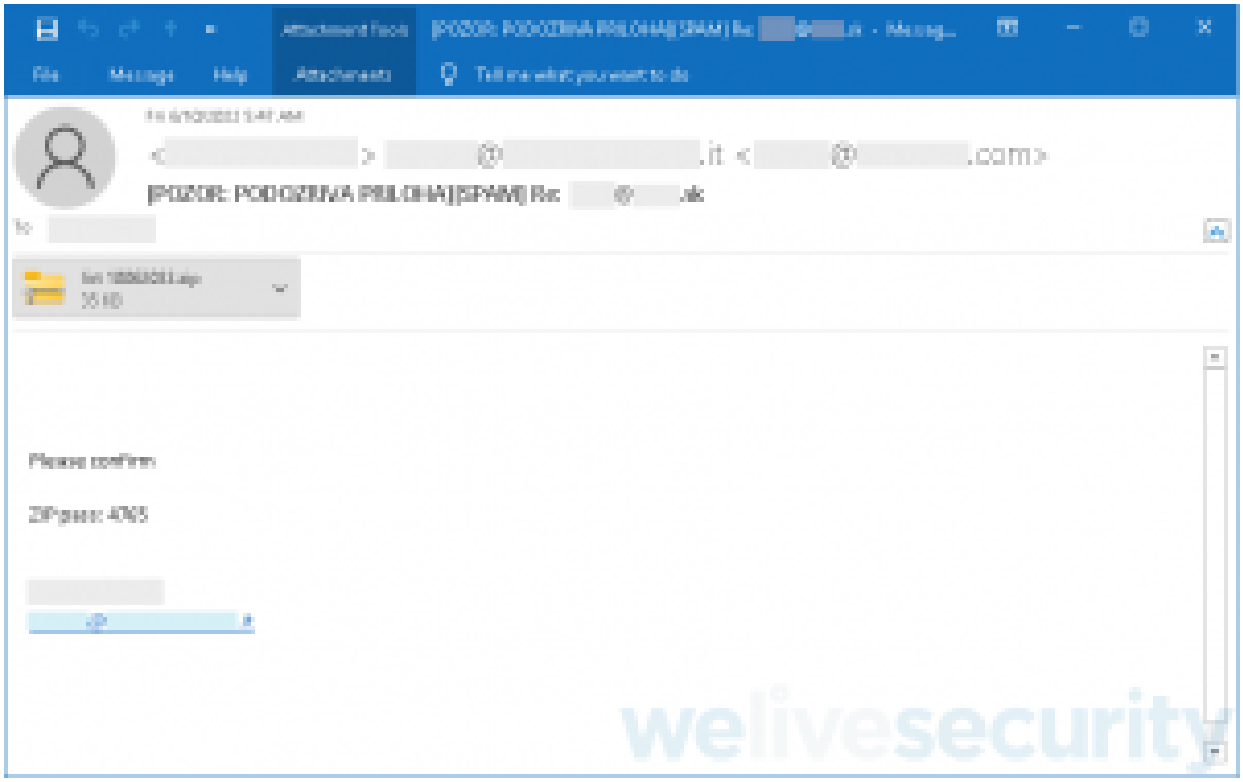


Figure 6. And another malicious email sent by Emotet's operators

When double-clicked, a shortcut file can launch a target resource, in this case, a PowerShell script that downloaded and executed Emotet:

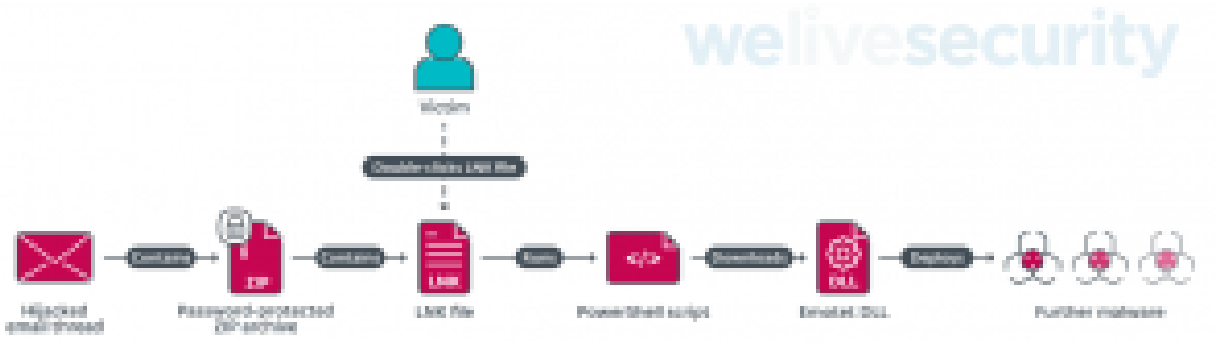


Figure 7. Emotet's operators use shortcut (LNK) files to deliver malware

Most detections were in Japan (28%), Italy (16%), and Mexico (11%).

In an earlier test campaign between April 4th and April 19th, the Emotet operators attracted victims to a ZIP archive, stored on OneDrive, containing Microsoft Excel Add-in (XLL) files, which are used to add custom functions to Excel. If extracted and executed, these files dropped and ran Emotet.

When Emotet's operators first resurrected their botnet from the takedown efforts in late 2021, another campaign was discovered that uses Cobalt Strike Beacon, a popular pentesting tool. By using a Beacon, the Emotet operators can decrease the time to deploy their final payload – often ransomware.

Mitigating macro malware

Emailing documents that contain macros is both a common occurrence in corporate environments and can serve as a technique to deliver malware when those macros are malicious. Recognizing this potential abuse of macros, during the heyday of Word 97 Microsoft introduced the first built-in security feature in Word that blocked Visual Basic for Applications (VBA) macros from running:

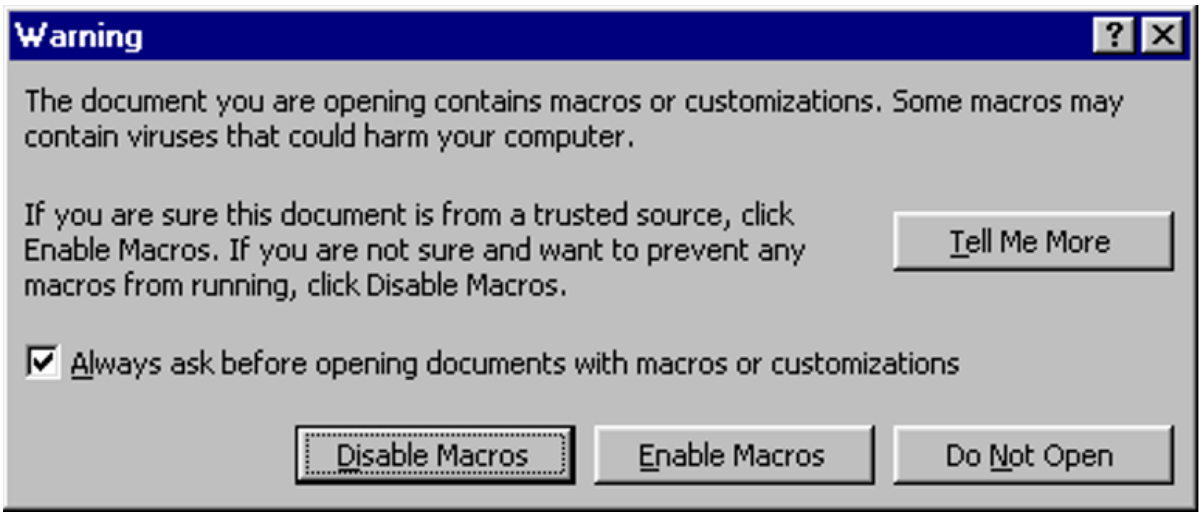


Figure 8. The default behavior of Word 97 when opening a document containing a VBA macro

This feature continued to be developed in later versions of Office, now probably most familiar via the yellow Message Bar with the “Enable Content” button introduced in Office 2010:

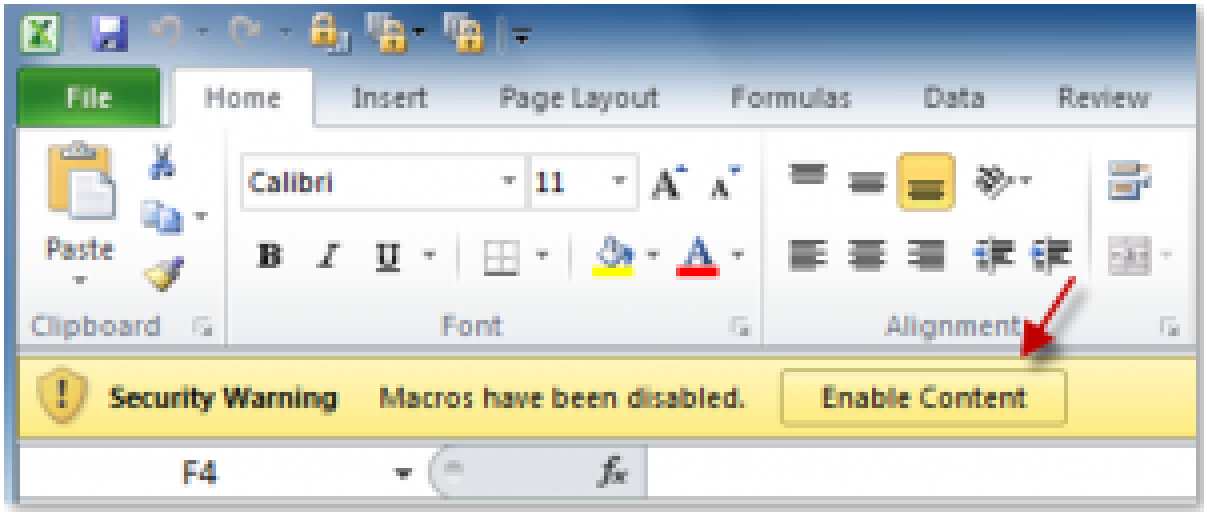


Figure 9. The Enable Content button in Excel 2010

Since then, two clicks have been typically required to enable macros: first, clicking on “Enable Editing”, which removes the document from Protected View, a security feature in place since Office 2010 that provides a read-only, sandboxed environment; second, clicking on “Enable Content”, which allows the macros to run. So long as an admin policy is not in place to prevent recipients from clicking through, the macros successfully load and run.

Although the blocking of macros helped limit the delivery of malware, malicious actors, such as the Emotet operators, adapted their efforts by focusing on duping victims into clicking through to enable macros.

With a phased rollout starting in April 2022, Microsoft has been tightening up the default handling of macro-enabled files downloaded from the internet by entirely removing the option to click “Enable Content”. After this change is deployed, macros are still blocked from running as before. So in order to run them, either the data about the file’s zone – sometimes called the Mark of the Web – needs to be removed, or the file has to come from a zone with a higher level of trust than that of the internet. These are much more complex actions to socially engineer recipients into and should thus help stymie future spam campaigns.

Since the increased security benefit offered by this change is only as strong as the Mark of the Web, let’s dive deeper into what it is, how it is used to determine when to block macros from running, and how spammers attempt to bypass it.

Deterring malware with the Mark of the Web

The Mark of the Web refers to the comment added to HTML files (as well as to MHT and XML files) indicating their host URL:

Figure 10. Browsers add the Mark of the Web to HTML files downloaded from the internet

This comment is automatically added by the Internet Explorer browser when the HTML file is being saved, or can be added manually by web site developers for testing or by other browsers and applications. The URL is then used to determine the level of trust assigned to the HTML file and any scripts or active content on which the URL might depend.

By default, every URL is treated as coming from the Internet zone: that is, as neither trusted nor untrusted. Although scripts and other active content embedded in the file can automatically run, they cannot access the local file system.

A URL could be added to different zones: the Restricted Sites zone for potentially unsafe content, or the Local Intranet and Trusted Sites zones for trusted content; there is also a Local Machine zone that, although it originally allowed scripts and ActiveX content to run automatically as trusted, was eventually equipped with a lockdown feature that prohibited such automatic execution.

Even before the demise of Internet Explorer, the Mark of the Web was also an informal name for the information about a file's zone that the New Technology File System (NTFS), the default file system of Windows NT-based operating systems, provides in an Alternate Data Stream (ADS). In NTFS, every file has an unnamed stream with a stream type of $DATA that contains the expected content of the file when it is opened by a program that can handle its file type:

Figure 11. Opening an NTFS file normally

Figure 12. The unnamed (default) stream of an NTFS file contains the same data as when the file is opened normally

The filename, the stream name, and the stream type are joined and delimited by colons. Thus, in the eyes of NTFS, filename.txt is equivalent to filename.txt::$DATA. Notice how there is no stream name, only a file name and a stream type.

On the other hand, the file's zone is contained in a stream that looks like this: filename.txt:Zone.Identifier:$DATA. The Zone.Identifier is a well-known stream name that modern browsers and some other applications automatically add or propagate to files to indicate their zone: the internet, the intranet, the trusted zone, the restricted zone, or the local machine. Some applications, such as the Chrome browser, add the host URL and the referrer URL to the Zone.Identifier as well:

Figure 13. Browsers can add the Zone.Identifier stream to files downloaded from the internet (ZoneId=3)

There are other known techniques to get around the Mark of the Web that the Emotet gang could try as well. It is possible to use container files, such as ISO disk images and VHDX files, or compressed/archive files, such as .arj and .gzip files, that do not propagate the Mark of the Web to files extracted from them. Ultimately, should one of these techniques yield a satisfactory return on investment, we can expect Emotet to return with force.

Abusing Alternate Data Streams

An NTFS file can contain an arbitrary number of streams, meaning these can and have been put to malicious use. For example, the Winnti Group operators stored a malicious, encrypted payload in a stream they named NULL.DAT. After decryption, the payload was either the PortReuse backdoor or the ShadowPad malware.

When the Turla operators deployed the Gazer backdoor against embassies and consulates around the world in 2016, the backdoor would hide its files in streams using GUIDs as stream names when it couldn't store them in the Windows registry.

Guildma also used streams as one method of hiding its binary modules, storing multiple files in the streams of a single file. Specifically, Guildma stored all of its malicious modules, including a couple of tools from Nirsoft for extracting saved credentials from popular email clients and web browsers, as the streams of the single desktop.ini file:

- desktop.ini:nauwuygiaa.jpg (MailPassView)
- desktop.ini:nauwuygiab.jpg (BrowserPassView)…

For targeting air-gapped networks, malicious actors have used streams to hide malicious components within otherwise innocuous-looking files on USB drives. The streams could contain data being stolen and command-and-control instructions from the malicious

operators. Considering that air-gapped networks lack an internet connection, clamping down on the use of USB devices and other portable storage devices travelling in and out of air-gapped networks is crucial for their continued security.

Some malware, like GoBotKR, can remove the Zone.Identifier stream from files to conceal the fact that they were downloaded from the internet zone. This entirely bypasses any protection that relies solely on the Mark of the Web to determine when to block macros from running.

Finally, spammers like the Emotet developers have taken a social engineering approach, attempting to trick recipients into enabling macros instead of removing the Zone.Identifier stream or using streams to hide payloads – until now.

Security tips

Be aware that some software does not add or propagate the Zone.Identifier stream, at least not consistently. For example, using 7-Zip to extract a .exe file from an archive downloaded from the internet does not propagate the archive's Zone.Identifier to its contents, meaning that there is no Mark of the Web to trigger any security blocks or warnings if any of the extracted files are run. The Zone.Identifier is propagated, however, by double-clicking on the .exe from within the archive. *(UPDATE: Version 22.00 of 7-Zip offers a new menu option and a new command line switch to enable propagation of the Zone.Identifier.)*

In light of the removal of the "Enable Content" button, a handy list that tracks whether file archivers support the Mark of the Web has been compiled in GitHub here.

For organizations that rely on macros as part of employees' workflows, IT admins may need to adjust the policies for how Office handles macros. Furthermore, organizations should take advantage of this opportunity to review their security stance against threats vectoring via email with the following:

- Use an email security solution that can block phishing, spam, and other malicious emails from reaching inboxes.
- Run phishing simulation exercises to test and renew employees' security awareness.
- Consider deploying a detection and response solution that can help track down whether the root cause of a cyberattack on your network was a malicious email or a different vector.

The impending close of the era of the "Enable Content" button has two consequences. First, that users can expect better protection against malicious macros delivered via email. Second, that spammers like Emotet are adapting their favorite tactics to dupe their future victims. Should any of these experiments prove successful, we can expect new malicious campaigns to hit inboxes, meaning that continued vigilance for email-based threats should remain top of mind.

UPDATE (June 27th, 2022): This article was updated to add information about new features in 7-Zip.

16 Jun 2022 - 11:30AM

Sign up to receive an email update whenever a new article is published in our Ukraine Crisis – Digital Security Resource Center

Newsletter

Discussion