# EmoCrash: Exploiting a Vulnerability in Emotet Malware for Defense

binarydefense.com/emocrash-exploiting-a-vulnerability-in-emotet-malware-for-defense/

August 14, 2020

**By: James Quinn**

Most of the vulnerabilities and exploits that you read about are good news for attackers and bad news for the rest of us. However, it's important to keep in mind that malware is software that can also have flaws. Just as attackers can exploit flaws in legitimate software to cause harm, defenders can also reverse-engineer malware to discover its vulnerabilities and then exploit those to defeat the malware. The difficulty, of course, is how to share the news about the vulnerability with other defenders while keeping it a secret from threat actors so they don't just patch the flaw. Researchers at Binary Defense have been doing exactly that since February 6th, and now it is time to share the details more publicly.

Emotet is a prolific and highly successful email-based malware, with a primary focus on email theft and loading additional malware as a service. Most commonly identified by its three distinct botnets and fairly obfuscated code flow, Emotet is a unique and persistent threat to organizations of all sizes.

Unique threats require unique solutions, which is why Binary Defense developed a killswitch exploiting a simple buffer overflow found in Emotet's installation process and shared it freely with the infosec community, avoiding public channels to ensure maximum uptime of the exploit before the threat actors behind Emotet patched their malware to close the vulnerability. This killswitch was alive between Feb 6th, 2020 – Aug 6th, 2020, or 182 days.


## Emotet's "New" persistence mechanism

In early February of this year, Emotet released a massive codebase overhaul, changing several of the installation and persistence mechanisms and introducing a polymorphic state-machine to their code flow. This added a layer of obfuscation to the loader, making analysis more difficult. One of the key changes was the removal of the word list and file generation algorithm used by Emotet during previous Emotet installs.

In its place was a new algorithm that generated a filename to save the malware on each victim system, using a randomly chosen exe or dll system filename from the system32 directory. This filename was encrypted (encoded) with an exclusive OR (XOR) key and saved into a registry value set to the victim's volume serial number. Additionally, the XOR key was set to the volume serial number in little endian format.

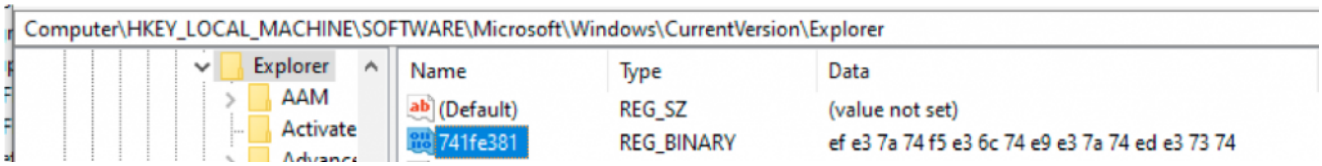


Fig 1, Emotet installed key

## Killswitch, V1

Around 37 hours after Emotet unveiled these changes, Binary Defense threat researcher James Quinn finished the first version of the killswitch/vaccine that eventually became EmoCrash. In the early days of this protection method, the PowerShell script would generate the registry key value for each victim and set the data for each value to null.

When Emotet would check Registry for the install marker, it would find the newly-generated null value and generate the exe name ".exe". It would then search the normal install location for this exe (%AppdataLocal%, C:\Windows\System32,C:\Windows\Syswow64, based on environment). If it didn't find it, it would drop a file to the normal install location called ".exe". However, when the malware attempts to execute ".exe", it would be unable to run because "." translates to the current working directory for many operating systems.

While this mechanism worked, it was very messy and still allowed Emotet to install—it just prevented Emotet from running successfully and reaching out over the network. Worried about lack of deployment due to the messiness of the protection mechanism, and the additional concern that it makes cleanup difficult, Binary Defense continued experimenting with the killswitch, looking for ways to increase its effectiveness.

## Killswitch, V2

Around 48 hours after Emotet released the newest loader version, Binary Defense completed the second version of the killswitch. This version exploited a simple buffer overflow discovered in Emotet's installation routine which caused Emotet to crash during malware install, but before the malware would drop itself to the normal Emotet install locations, thus completely preventing malware installation. Additionally, two crash logs would appear with event ID 1000 and 1001, which could be used to identify endpoints with disabled and dead Emotet binaries after deployment of the killswitch (and a computer restart).

```
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Application Error"/>
    <EventID Qualifiers="0">1000</EventID>
    <Level>2</Level>
    <Task>100</Task>
    <Keywords>0x80000000000000</Keywords>
    <TimeCreated SystemTime="2020-02-07T04:07:35.244535200Z"/>
    <EventRecordID>622</EventRecordID>
    <Channel>Application</Channel>
    <Computer>DESKTOP-BB23TQ0</Computer>
    <Security/>
  </System>
  <EventData>
    <Data>sqlsrv32.exe</Data>
    <Data>0.0.0.0</Data>
    <Data>5e3c3d86</Data>
    <Data>ntdll.dll</Data>
    <Data>10.0.18362.1</Data>
    <Data>9bbcb4a9</Data>
    <Data>c0000374</Data>
    <Data>000df8cd</Data>
    <Data>370</Data>
    <Data>01d5dd6c1f53dab3</Data>
    <Data>C:\Windows\SysWOW64\sqlsrv32\sqlsrv32.exe</Data>
    <Data>C:\Windows\SYSTEM32\ntdll.dll</Data>
    <Data>f0e9a351-a3b7-41c2-b3fe-741d993363a7</Data>
    <Data/>
    <Data/>
  </EventData>
</Event>
```

Packaged into a nice usable PowerShell script (which we named EmoCrash), this utility would identify the user's architecture and then generate the corresponding install registry value for the user's volume serial number. It would then generate a buffer of 0x340 (832) bytes, which it would save to the new registry value's data.

This tiny data buffer was all that was needed to crash Emotet, and could even be deployed prior to infection (like a vaccine) or mid-infection (like a killswitch).

## Exploit Release

With an incredible amount of coordinating between the Infosec and CERT communities, especially those at Team Cymru who helped immensely with this, Binary Defense began distributing the EmoCrash exploit script to defenders around the world on Feb 12, 2020, with strict instructions not to post it publicly. As our goal was to maximize the amount of good

done by this exploit while also avoiding tipping off the threat actors and allowing them to patch their code, Binary Defense set the information disclosure rules to TLP:Green, which limits disclosure to non-public channels.

As we began disclosing the exploit to the community, we received a lot of helpful feedback, including a fix to some compatibility issues that arose with Windows 7 deployments, contributed by the CERT at the University of Frankfurt in Germany. We would like to offer our most sincere thanks and appreciation to the members of the information security community around the world who worked to help keep this killswitch a secret from attackers while widely distributing it to defenders.

## Emotet Enters “dev mode”

Around Feb 7th, Emotet entered a period of time where they stopped spamming and began working on developing their malware. This time period stretched from Feb 7 – July 17, 2020. While their spam distribution was completely halted, they were not “inactive” during this time, as they continued to push out a few core binary updates and protocol updates. Additionally, Emotet was observed using Trickbot for loads as early as April 2, 2020.


## April 2020 Update

In mid April, Emotet released a new protocol change, along with changes to the core binary. One of the key changes was the introduction of a new installation method, retiring the Explorer registry key method. Additionally, this new update removed the use of concurrency mutexes, possibly in a bid to evade other more public killswitches.

While the Explorer registry key install method was retired, it was not removed entirely from the code. Instead, the explorer registry key was accessed by the malware when it checked to see if there were old installs that needed removing.

When it accessed this key, EmoCrash still triggered and would crash the malware before it could communicate out. Unfortunately, this crash would trigger after Emotet had installed itself (including service creation and file drop). However, the malware would not connect out and the service would not run. Additionally, as the crash still triggered event ID 1000 and 1001, responders were able to easily locate and remove these files.

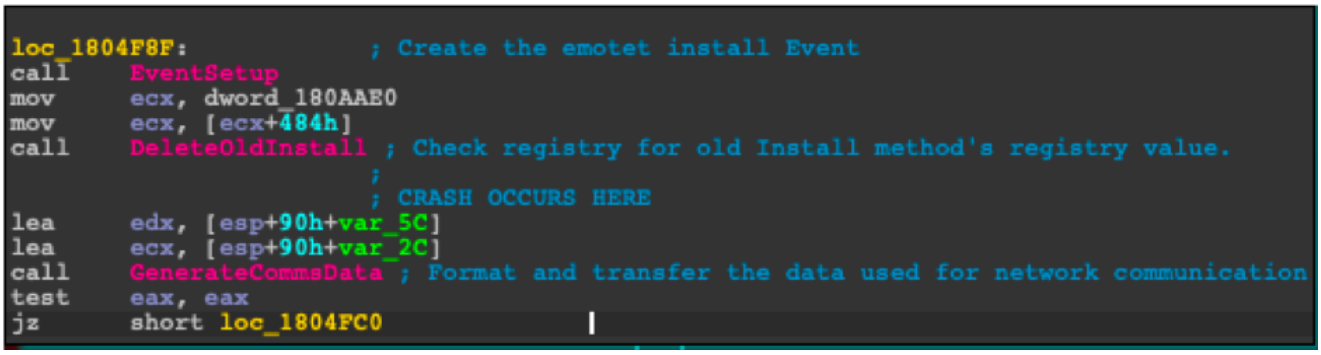


Fig 2, Conveniently, Emotet would crash right before data stolen was formatted and sent to the C2

## May 2020 – July 17, 2020

While Emotet did not return in force until July 17, 2020, Binary Defense received notifications from various organizations that had deployed EmoCrash and found and mitigated an ongoing Emotet infection. As Emotet was not spamming during this time, the loss of entire victim organizations had to have a negative impact on the botnet's operations.

## July 17 – August 6, 2020

On July 17, 2020, Emotet finally returned to spamming after their several months-long development period. With EmoCrash still active at the start of their full return, up until August 6, EmoCrash was able to provide total protection from Emotet.

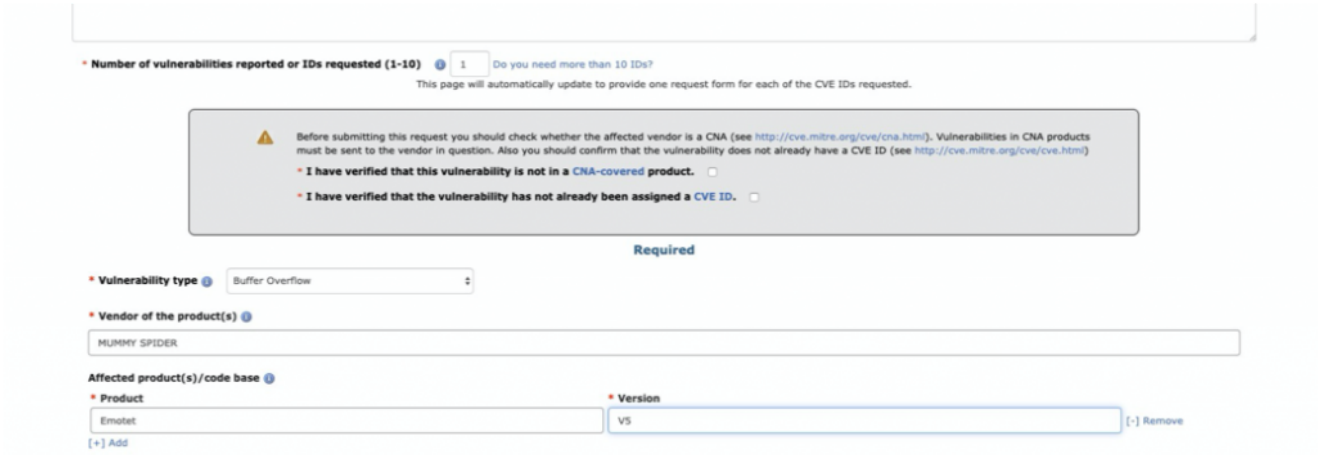
Not bad for a 832-byte buffer!

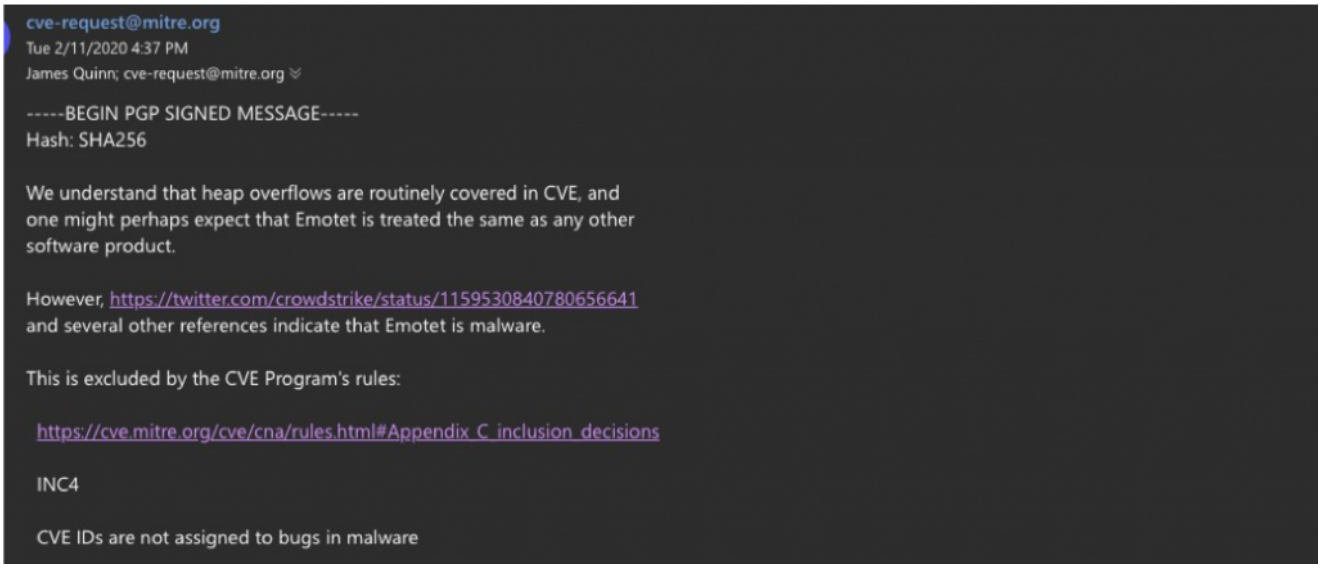
On August 6th, a core loader update was sent out which finally removed the vulnerable registry value code, effectively "killing" EmoCrash.

## Thank you to the security community

As stated above, we'd like to offer our sincerest thanks to everyone who worked hard to keep this exploit alive as long as possible. A huge thank you to Team Cymru for all the work they did to spread this throughout the community (while also keeping it off public channels).

Just for fun, I submitted a vulnerability report to MITRE's CVE program to see if they would assign it a CVE number. It is for the best that it was denied, since public disclosure would have been exactly the opposite of what we wanted, but it was fun to get a denial from MITRE about it.


* Number of vulnerabilities reported or IDs requested (1-10) 1 Do you need more than 10 IDs?
This page will automatically update to provide one request form for each of the CVE IDs requested.
Before submitting this request you should check whether the affected vendor is a CNA (see http://cve.mitre.org/cve/cna.html). Vulnerabilities in CNA products must be sent to the vendor in question. Also you should confirm that the vulnerability does not already have a CVE ID (see http://cve.mitre.org/cve/cve.html)
* I have verified that this vulnerability is not in a CNA-covered product.
* I have verified that the vulnerability has not already been assigned a CVE ID.
Required
* Vulnerability type Buffer Overflow
* Vendor of the product(s)
MUMMY SPIDER
Affected product(s)/code base
* Product
* Version
Emotet
V5
[-] Remove
[+] Add


cve-request@mitre.org
Tue 2/11/2020 4:37 PM
James Quinn; cve-request@mitre.org
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256
We understand that heap overflows are routinely covered in CVE, and one might perhaps expect that Emotet is treated the same as any other software product.
However, https://twitter.com/crowdstrike/status/1159530840780656641 and several other references indicate that Emotet is malware.
This is excluded by the CVE Program's rules:
https://cve.mitre.org/cve/cna/rules.html#Appendix_C_inclusion_decisions
INC4
CVE IDs are not assigned to bugs in malware