Hunting Emotet with Brim and Zeek

medium.com/brim-securitys-knowledge-funnel/hunting-emotet-with-brim-and-zeek-1000c2f5c1ff

Oliver Rochford December 8, 2020

Oliver Rochford

Nov 5, 2020

.

8 min read

The US Cybersecurity and Infrastructure Security Agency recently released an advisory warning of a resurgence of the Emotet malware.

Emotet started out in 2014 as a Banking Trojan, but has since evolved into a sophisticated malware, offered on the Darknet as a commercial Cybercrime-as-a-Service platform.

Victims that are infected with Emotet are usually targeted with a phishing email containing a macro-enabled malicious document, or a link to one hosted on a compromised website. The malware frequently acts as a "dropper" and downloads additional components and payloads. Emotet has worming capabilities and may attempt to enumerate and infect further victims on accessible networks. Command and Control (C2) is executed via HTTP POST requests on ports 80, 443 and 8080 to randomized alphanumerical named directories on compromised C2 servers.

What makes Emotet really dangerous is that it is sold as an operational platform to a variety of different threat actors with diverse motivations. Further, the malware can deploy different payloads depending on the objective — from stealing banking credentials to ransomware. It is essentially infrastructure-as-a-service for hacking. In practise this means that it constantly evolves and can come in many guises.

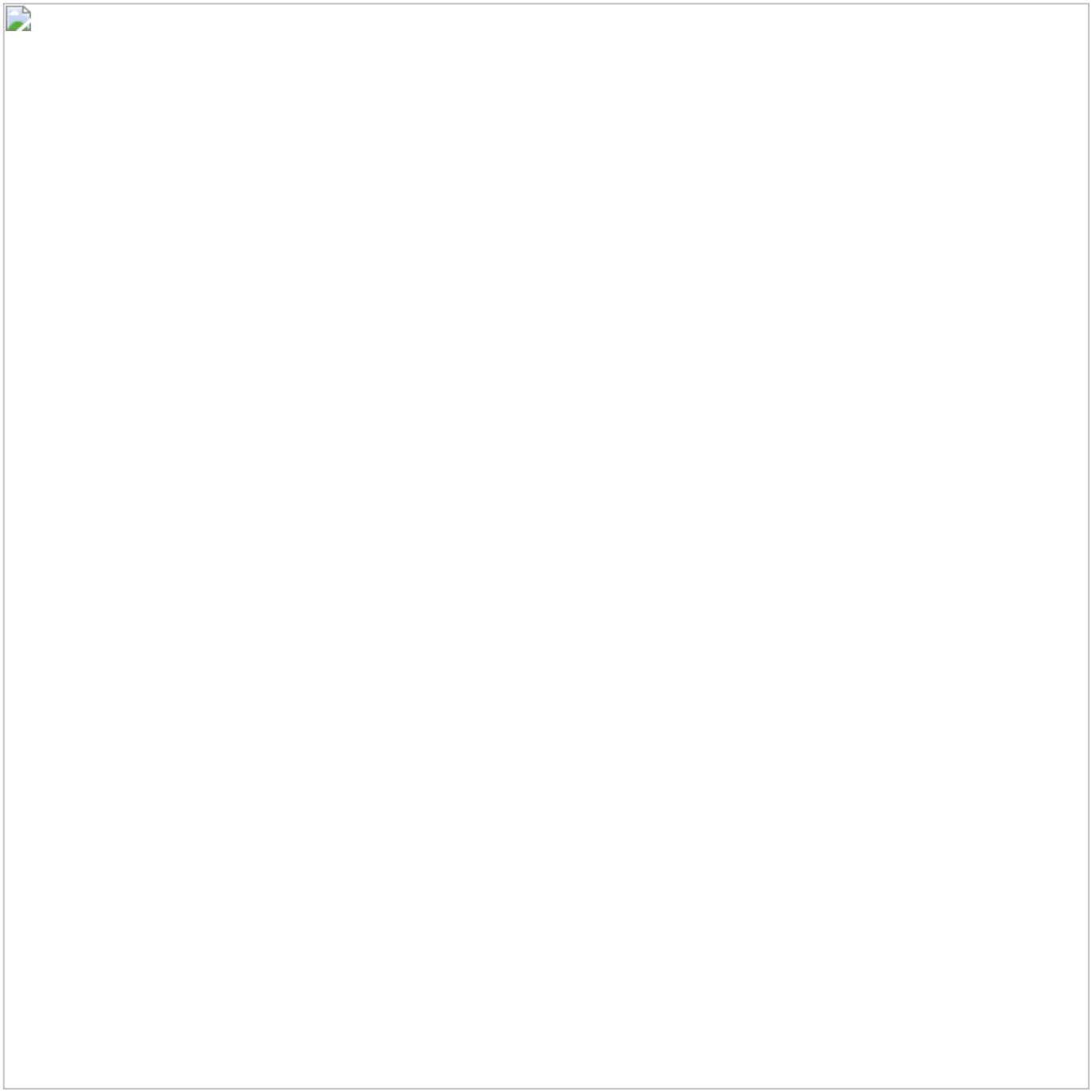
With Emotet on the rise again, blue team and incident response teams should familiarise themselves with how this dangerous threat behaves and evaluate how best to detect and hunt it. Luckily, there are samples available. In this article, we are specifically working with the following sample from the good people at Malware Traffic Analysis:

We'll also be using the Brim Desktop client. Let's go hunt!

> **TIP!** You can find detailed installation instructions for Brim on Windows, Linux and macOS under https://github.com/brimsec/brim/wiki/Installation

Finding Patient X

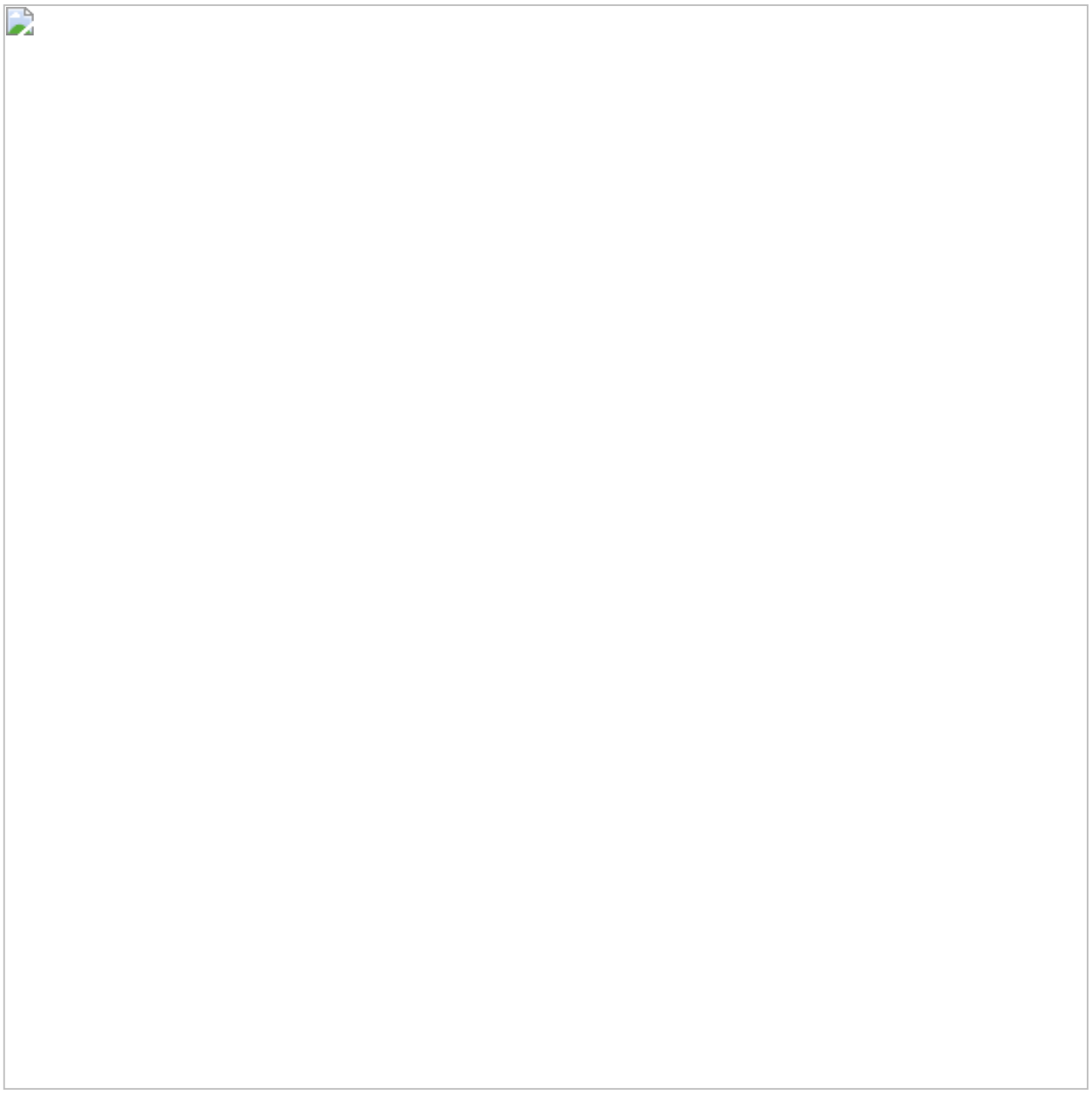
As we discussed in the first article, a good first step when investigating malware is to investigate the DNS activity, specifically for which domains there are resolution requests. Zeek provides a DNS protocol analyzer specifically for this purpose.



Zeek's DNS stream in Brim

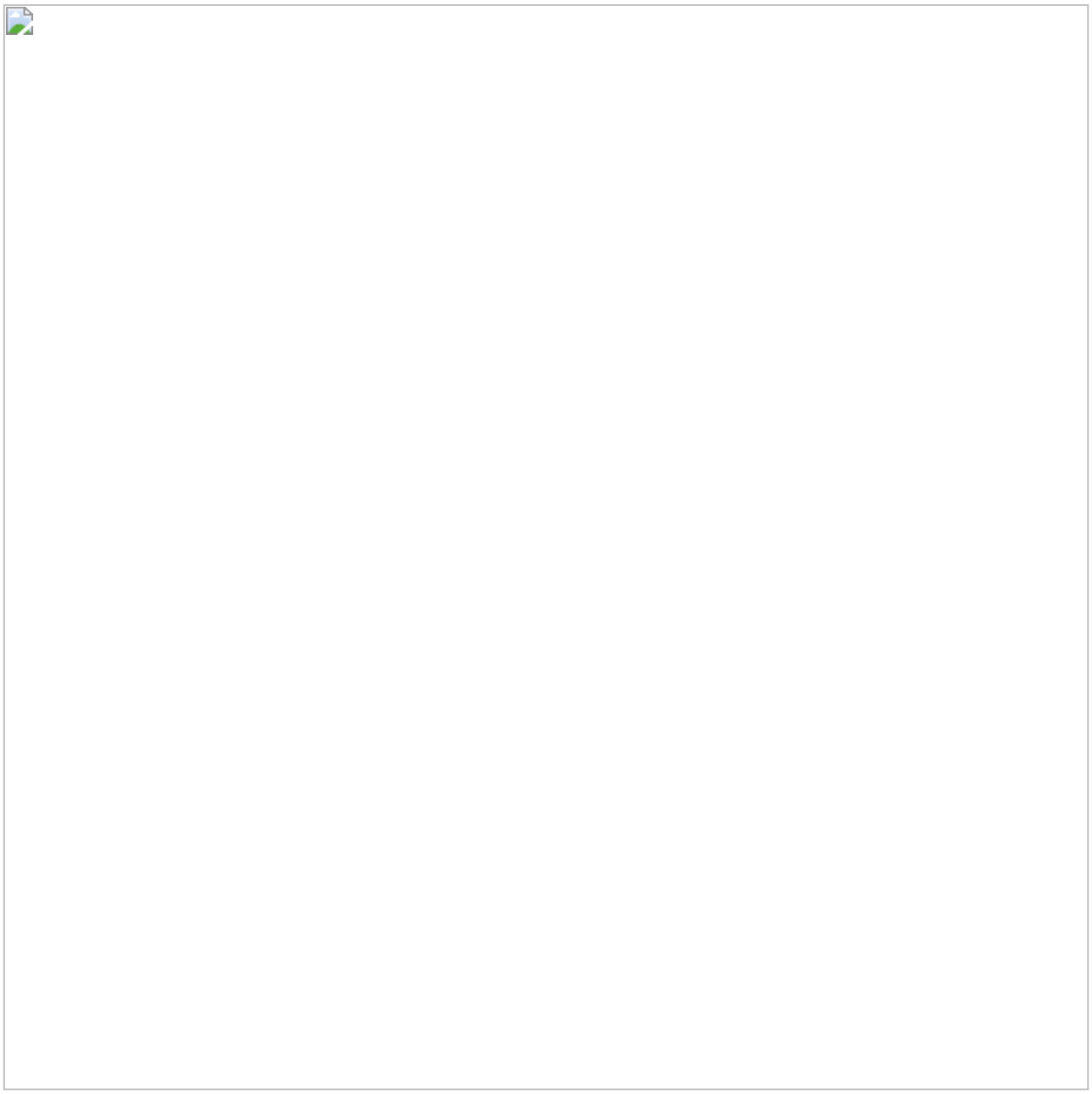
To gain an overview of what's going on, we'll use a ZQL query to stack the queries by count

```
_path=dns | count() by query | sort -r
```



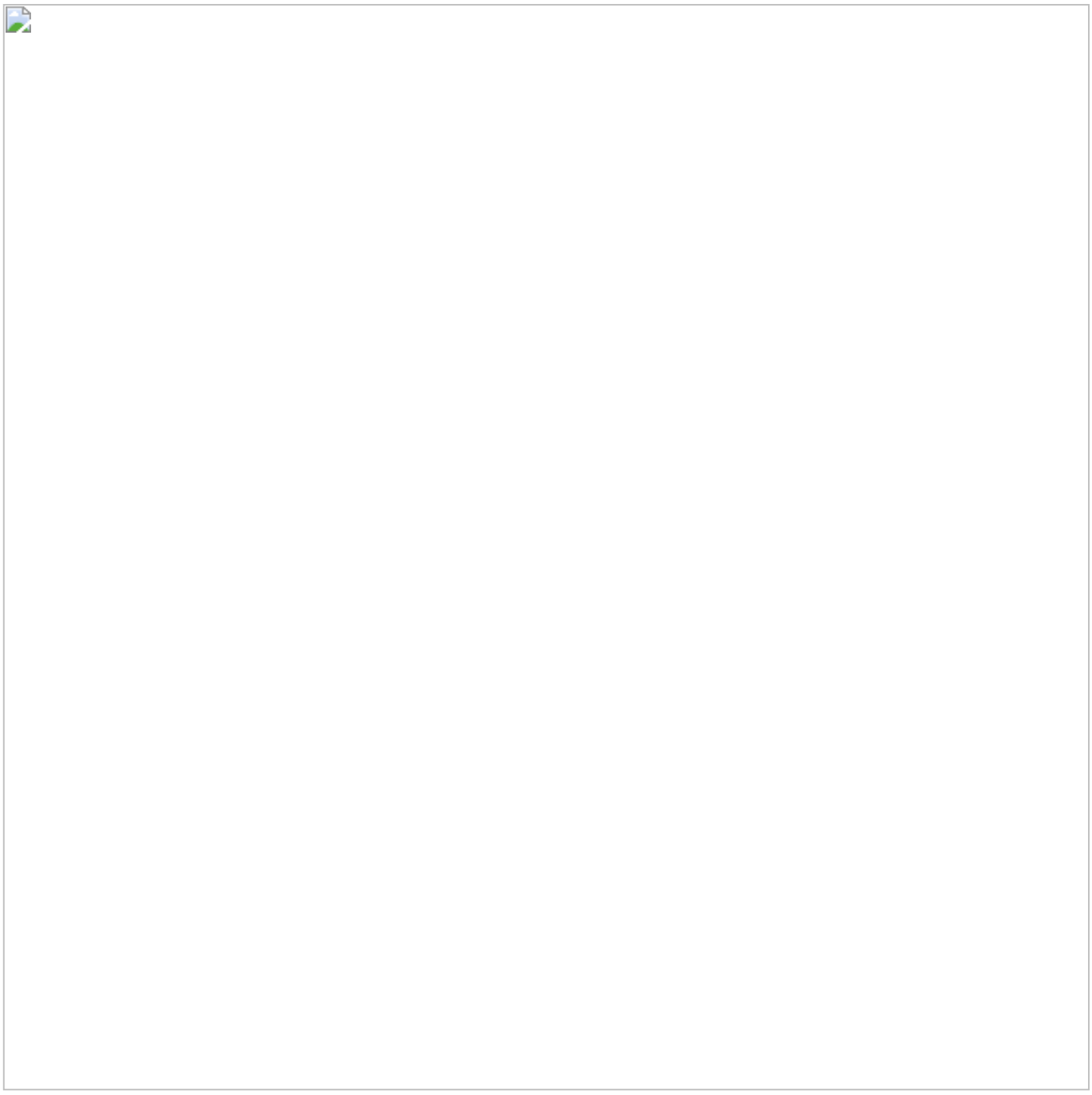
Count of unique DNS queries

We see a number of different windows network requests, for example for "*WORKGROUP*", some legitimate "*Microsoft.com*" requests, and single requests to threat-intelligence related addresses such as for *Spamhaus*. We can right-click on any domain we do not recognize and verify them using VirusTotal.

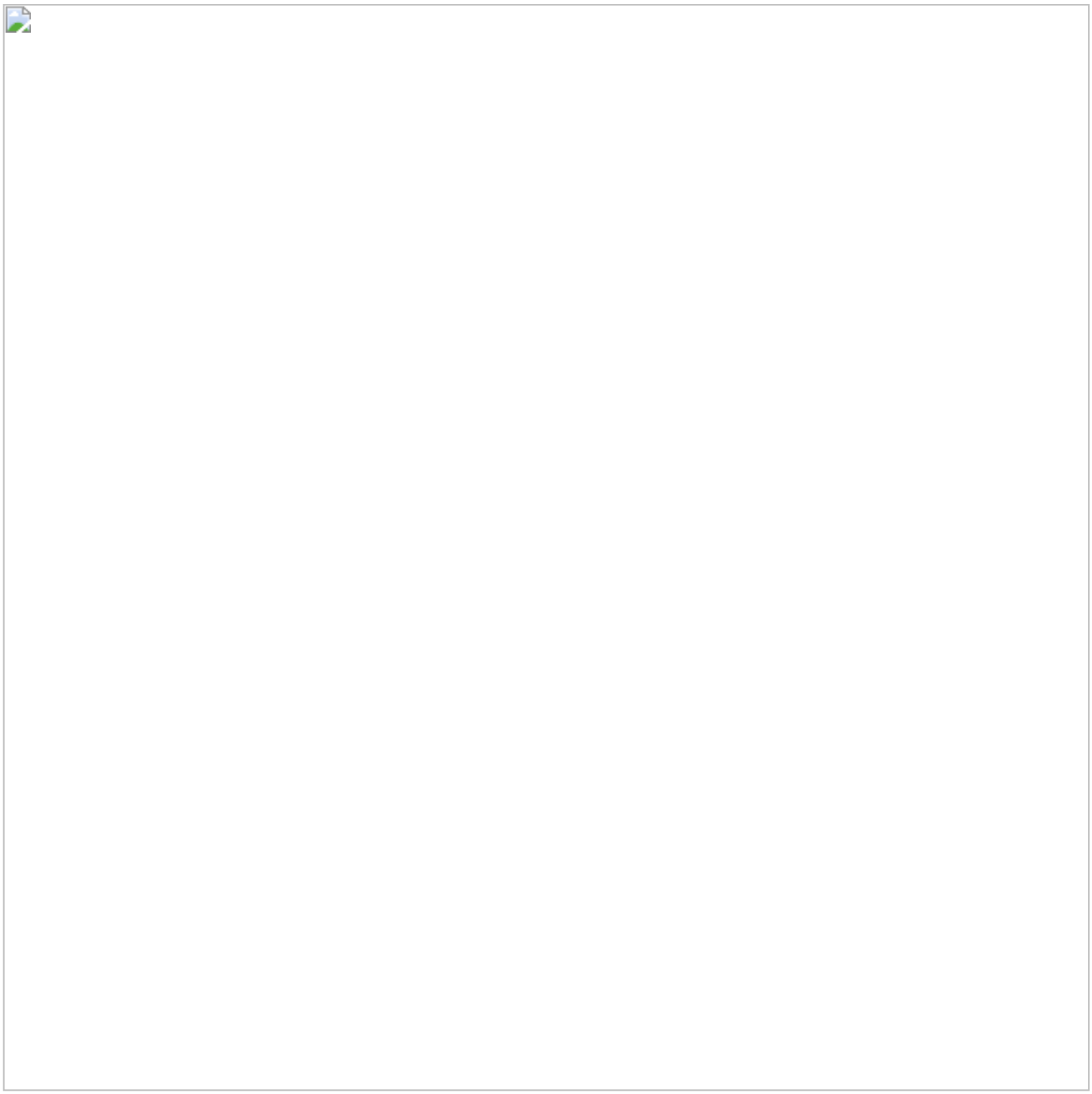


Right-click on any domain to validate it using VirusTotal

VirusTotal flags one of the domains, “ *t-privat.de*”, as malicious, and known to have hosted malware in the past. We now have a trail to follow.

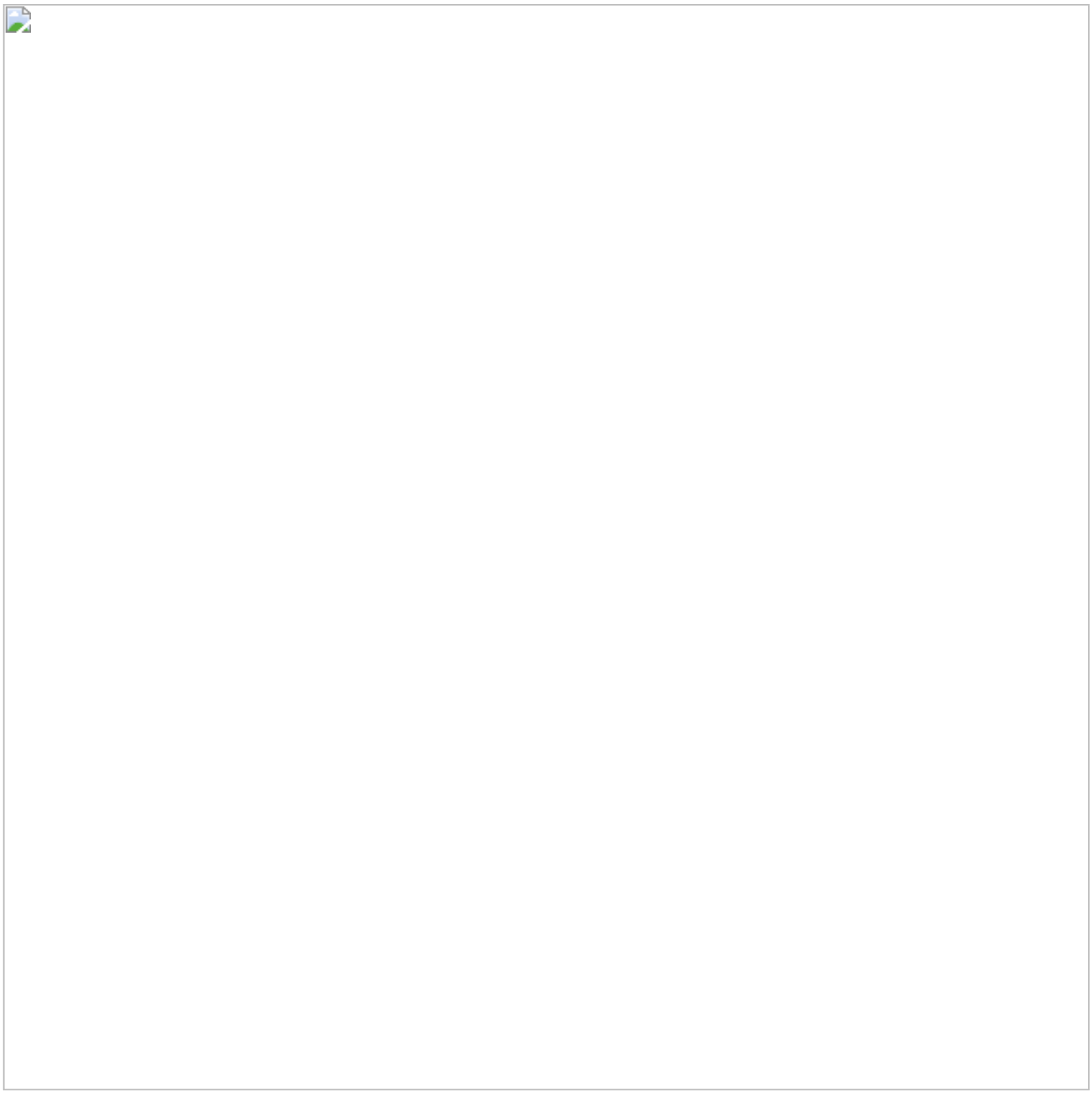


VirusTotal shows the suspicious URI as malicious



Details for the malicious Domain

A nice little blind check at this point is to verify if the relevant Host A Record IP address "*81.169.145.161*" is present anywhere else in our dataset.

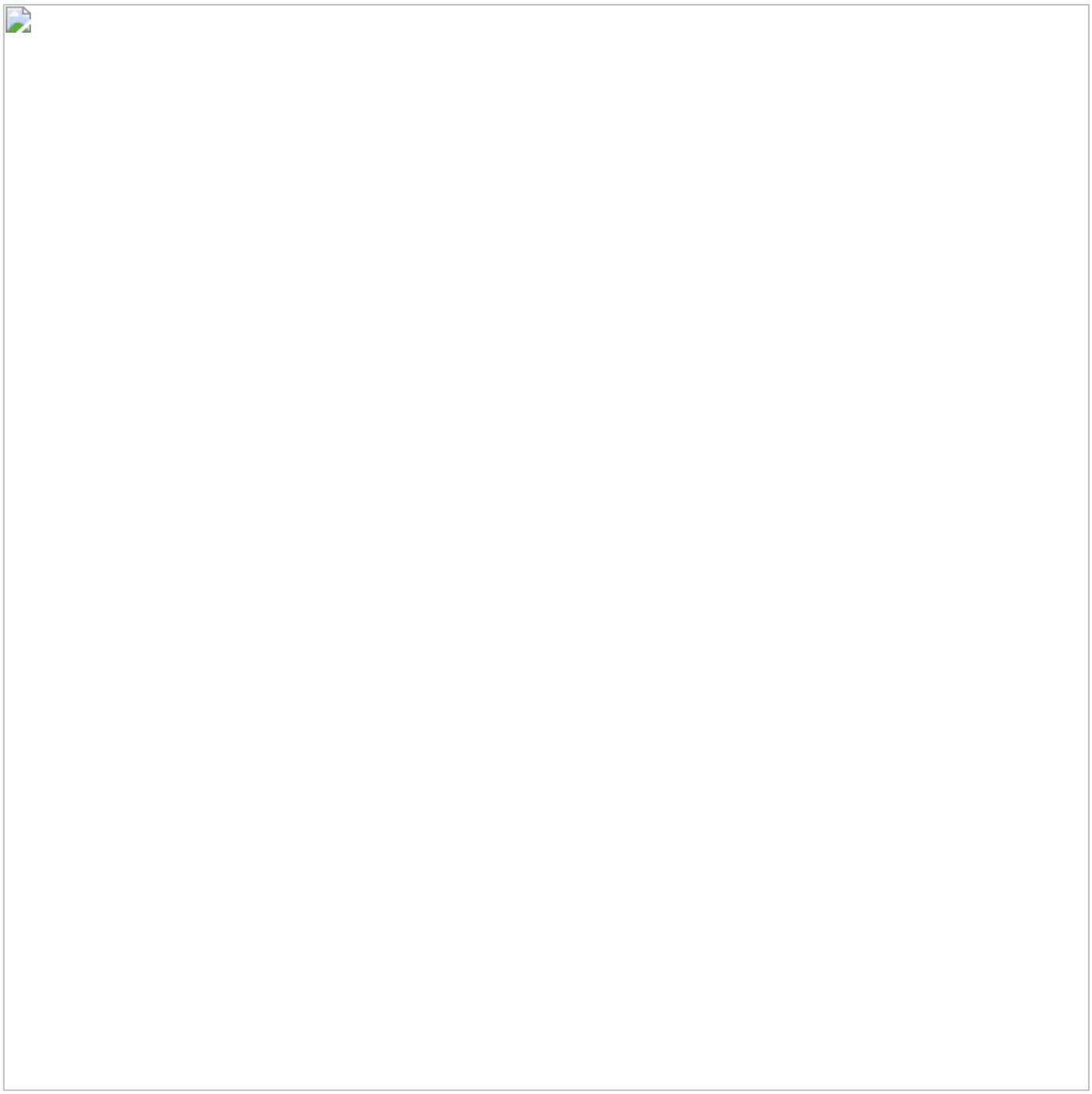


Hit! The IP address related to the VirusTotal alert matches our data

The search returns positive — we are looking at the same threat that was submitted to VirusTotal. Note the file transfer via HTTP to an internal host. That's something we want to investigate further.

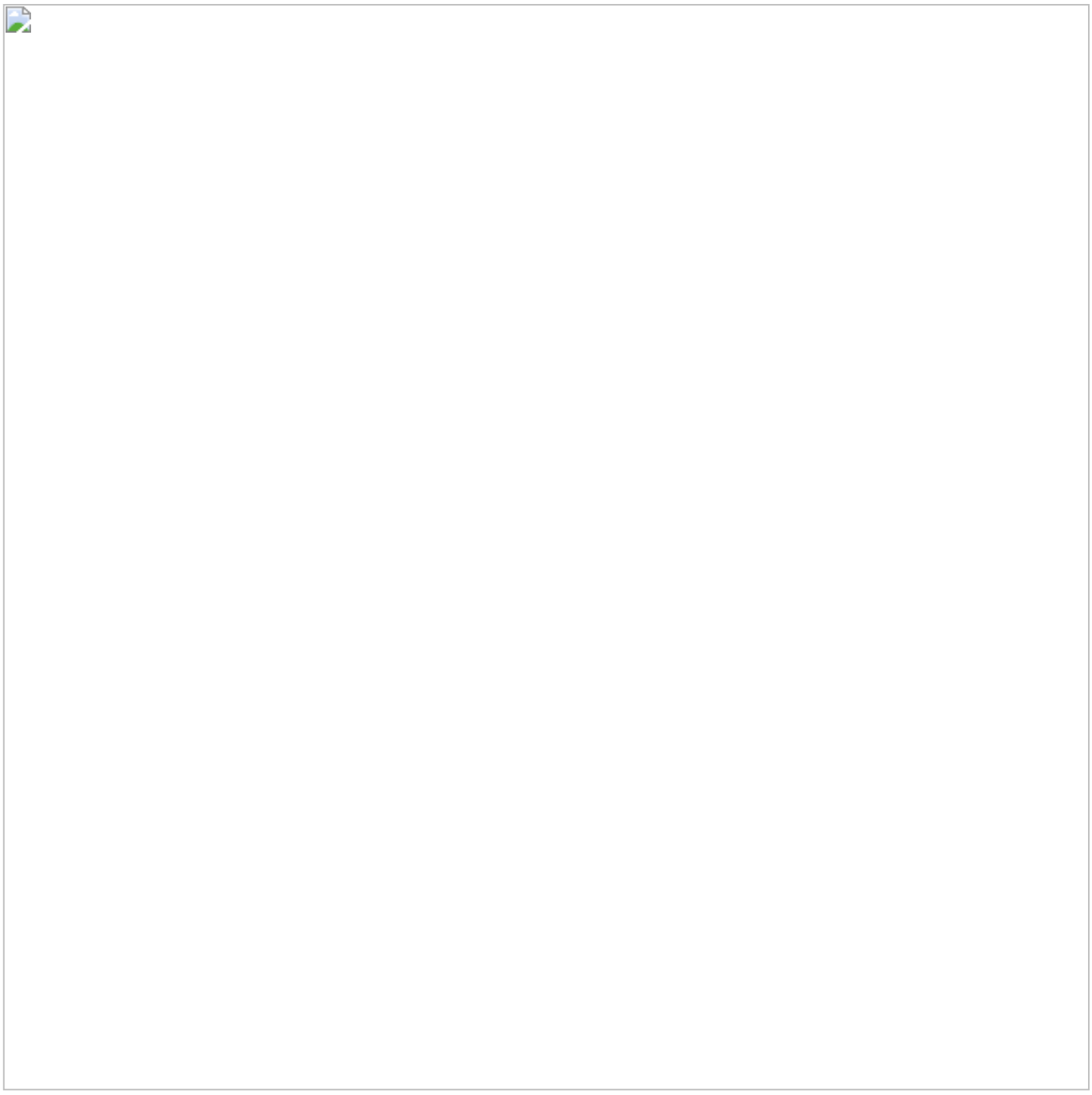
We'll revisit one of our Power Queries to show all of the relevant file activity within our data that has a complete filename, but extend it to provide us with a tailored view showing us only the fields that are relevant to us at this juncture:

```
filename!=null | cut _path, tx_hosts, rx_hosts, conn_uids, mime_type, filename, md5,
sha1
```



Curated File Activity view — the Plot thickens

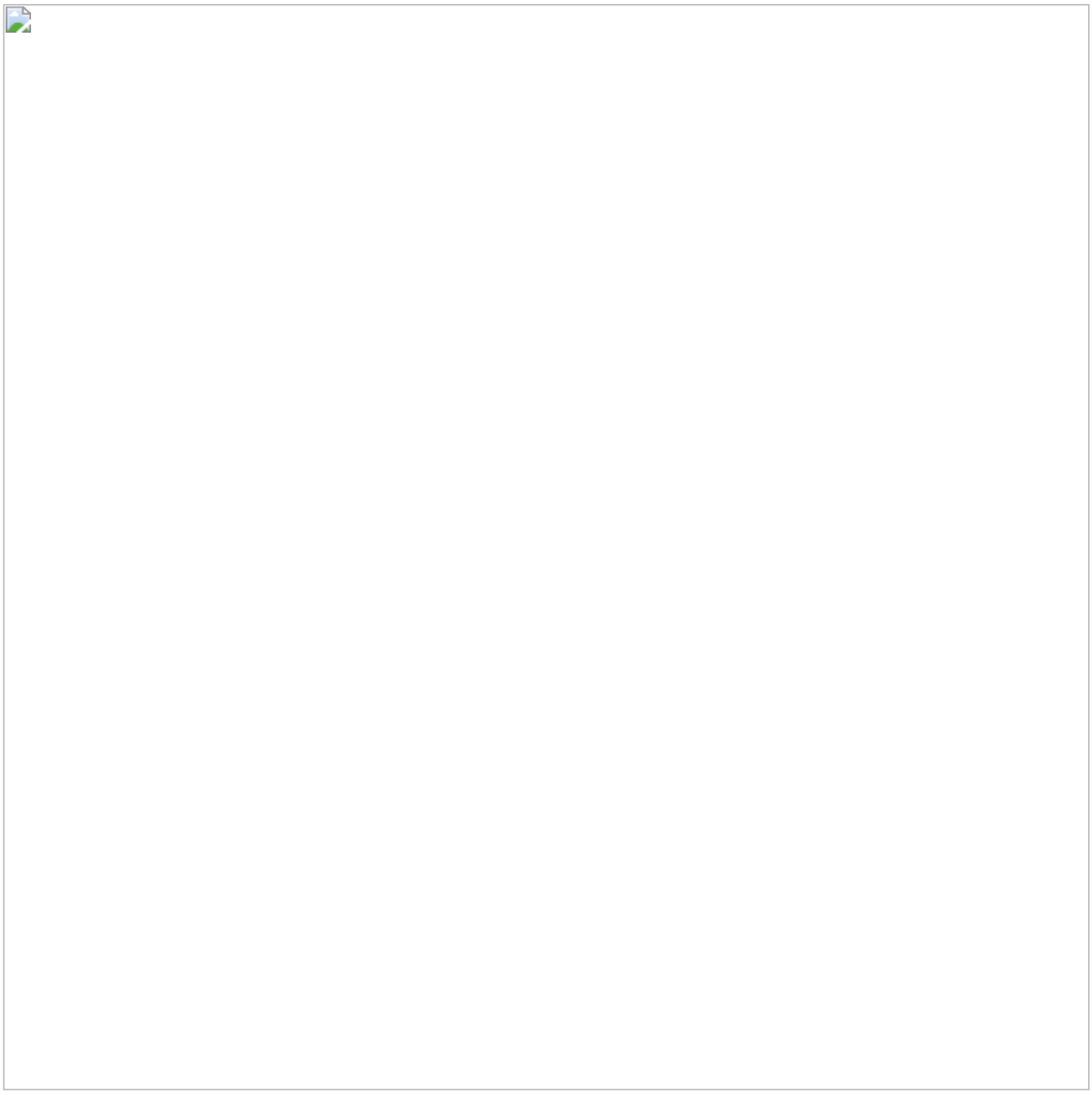
There are a number of strange looking HTTP connections with filenames lacking any type of extension or MIME type. We also find the full filename, “UR608.exe”, from our malicious internet host “*81.169.145.161*” (*t-privat.de*). Thanks to Zeek’s MD5 and SHA1 processors, we can harness VirusTotal again. Our Patient X is “*10.9.1.101*”.



Emotet sighted! VirusTotal confirms the file is associated with known malware.

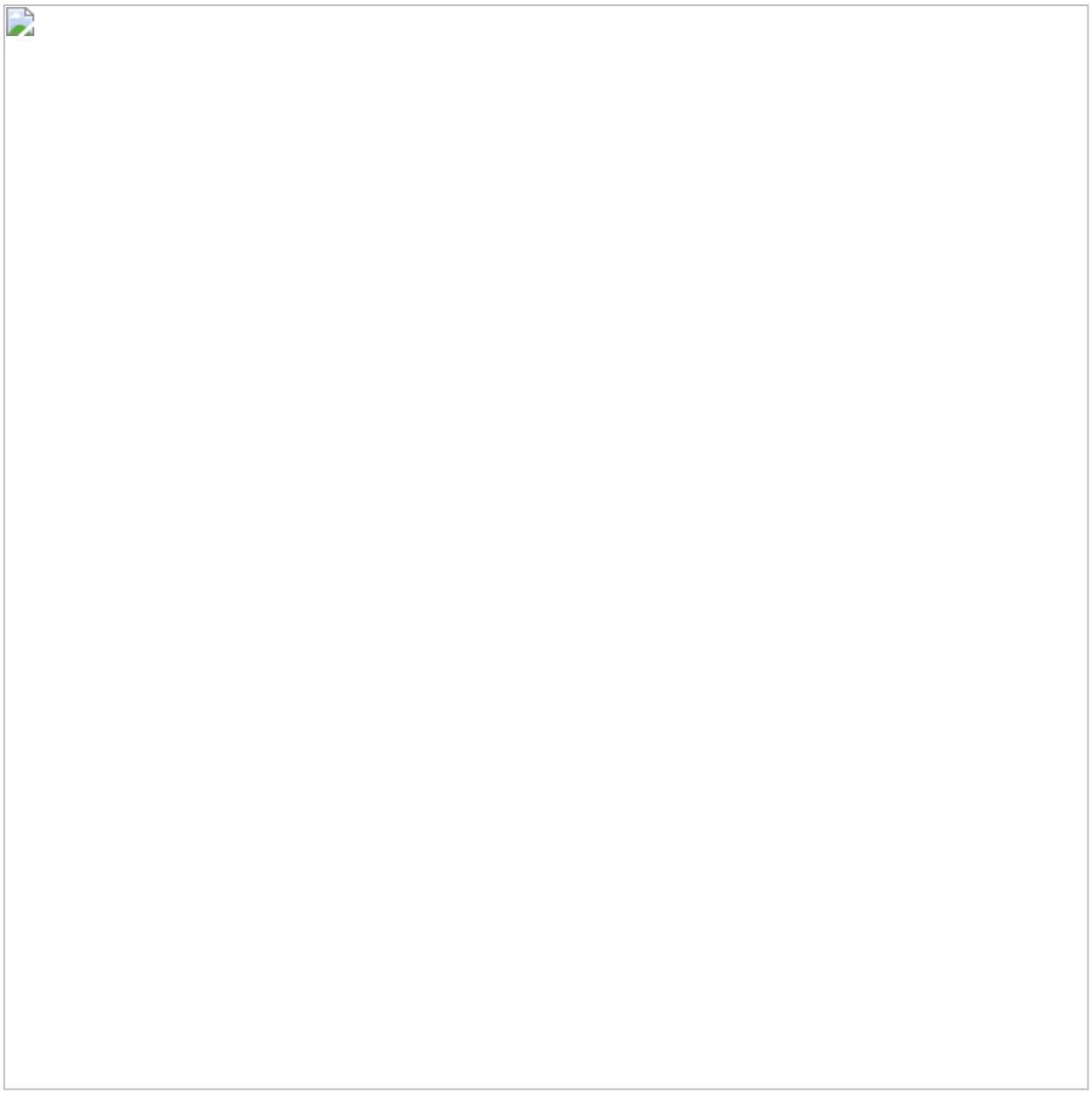
Identifying Command and Control

Now that we've verified the initial point of access for the attack, we can follow how the attack unfolded. When we reviewed the files contained in our data, we saw three other connections originating from the infected system. If we double-click on any of the records in question, we conjure up a detailed view showing a waterfall that includes all app transactions and file payloads throughout the life of a flow:

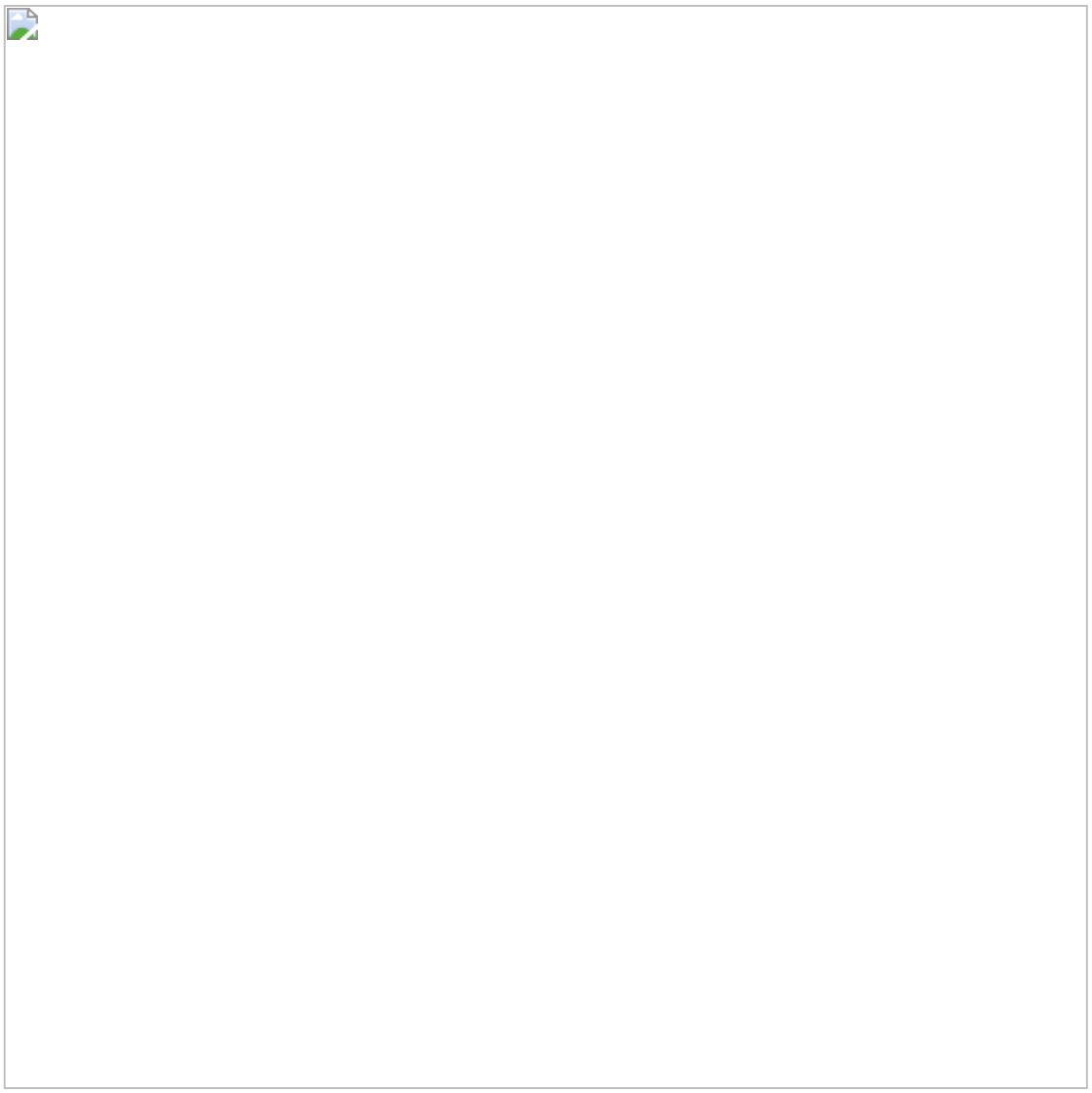


Connection Diagram for the suspicious HTTP requests from our Patient X

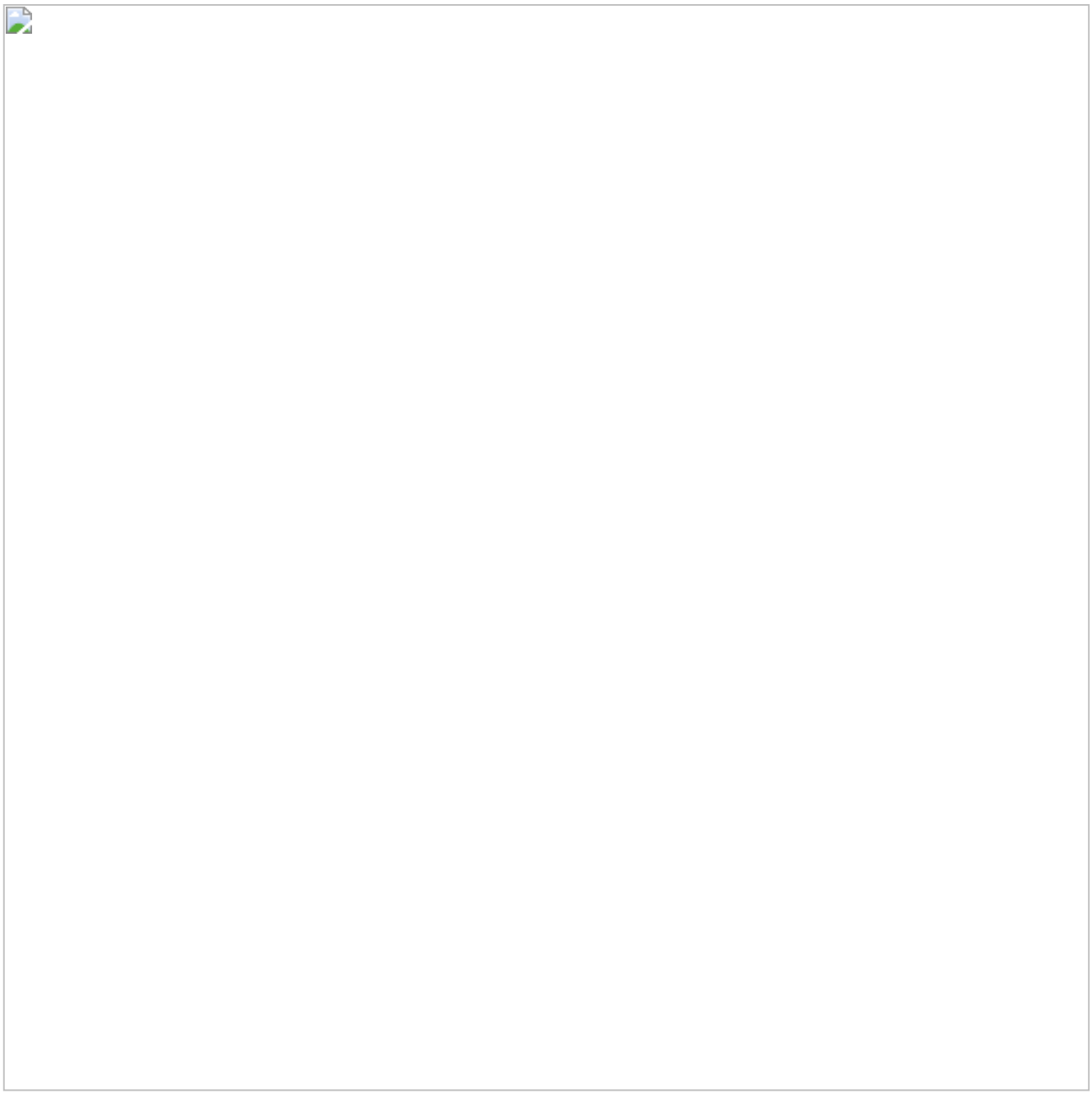
We can see a connection over a three minute time period. If you're familiar with Zeek, the "dpd" and "weird" events immediately stick out. Let's look into these. You can pivot right into the relevant details by clicking on any entry in the Connection Diagram. Zeek's "Dynamic Protocol Detection" (dpd) stream processor detects network protocol anomalies. The "weird" processor on the other hand alerts on unusual activity such as malformed requests.



Brim’s detailed view for Zeek’s “dpd” stream

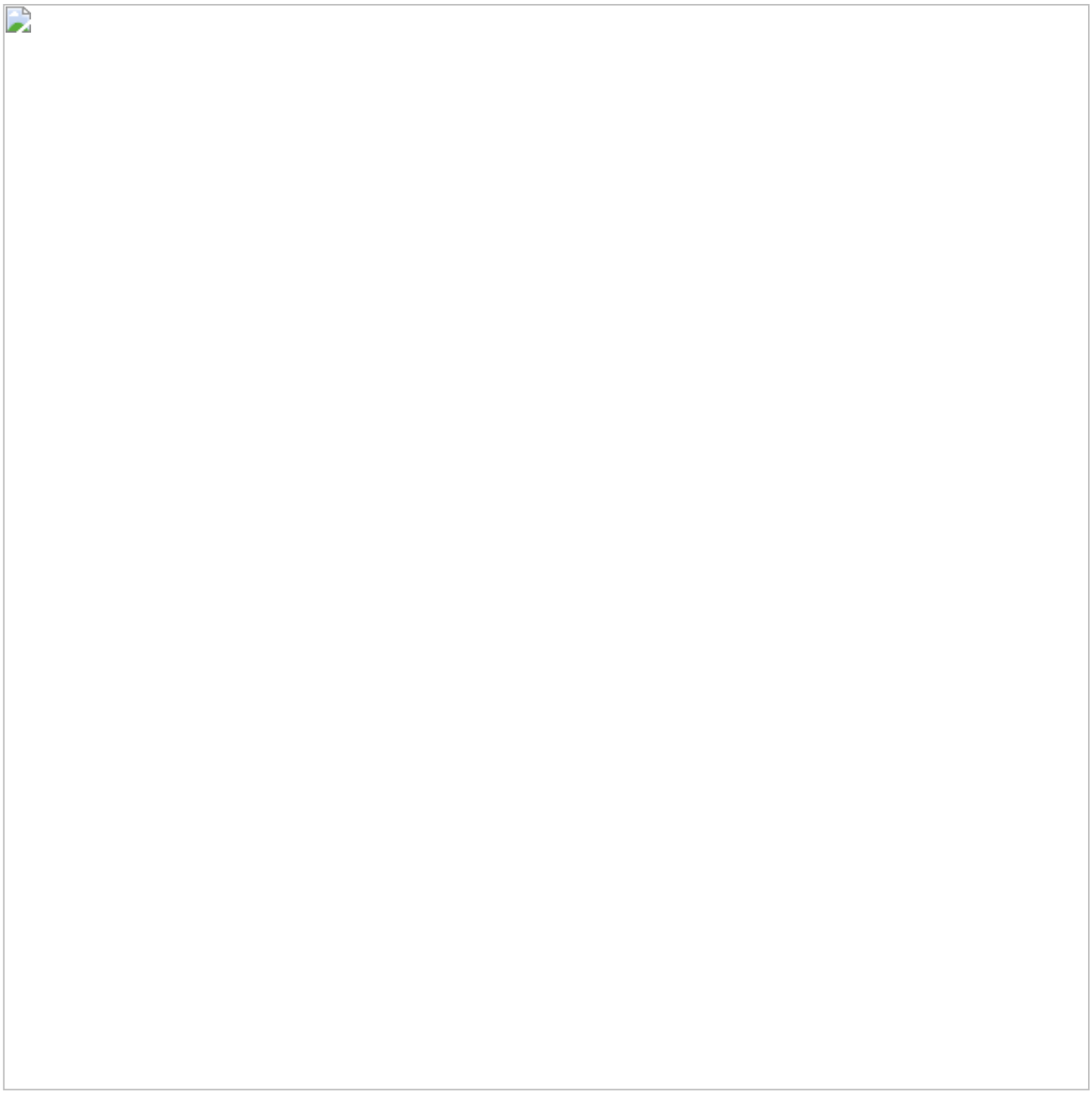


Brim’s detailed view for Zeek’s “weird” stream



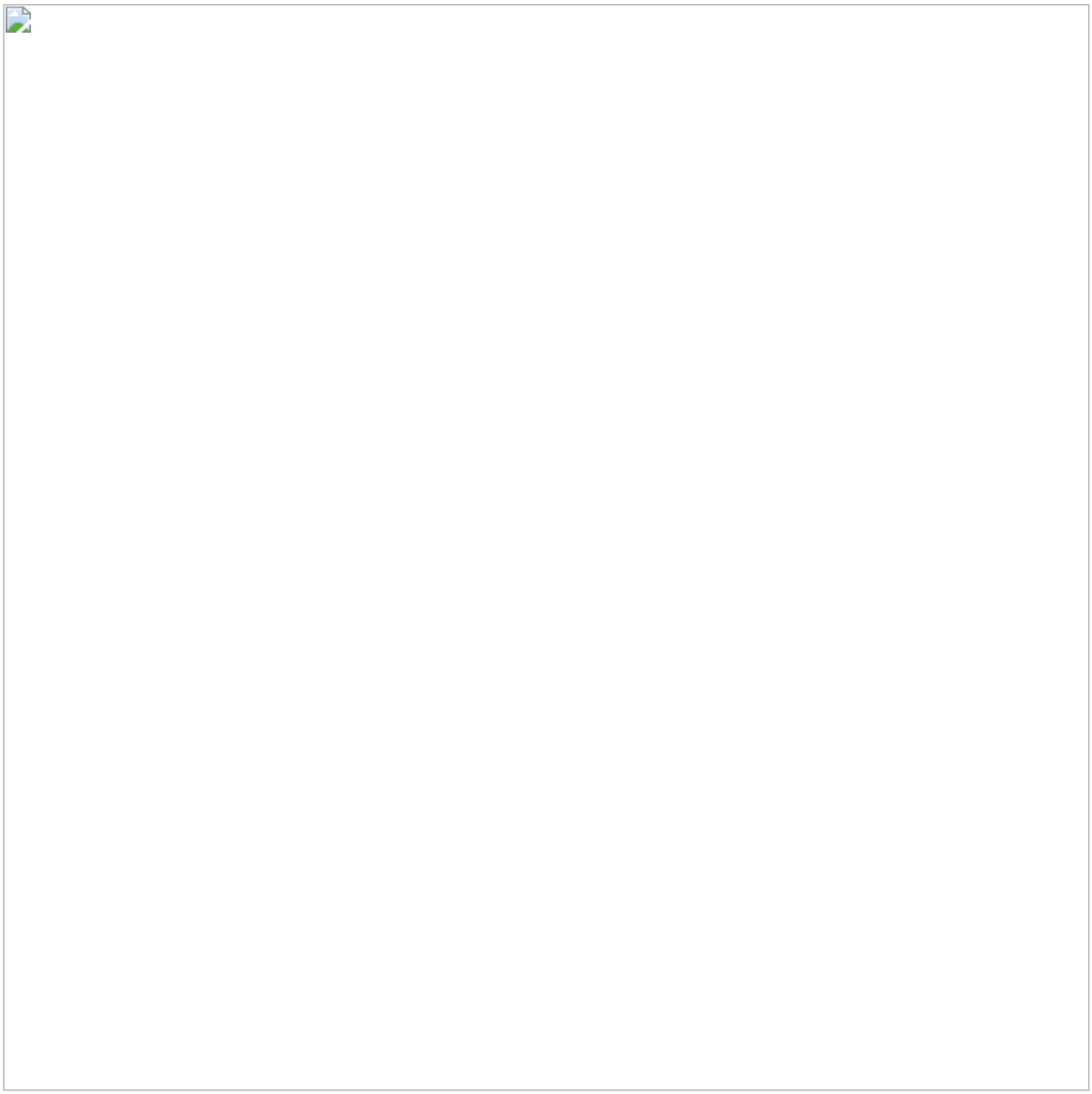
Brim’s detailed view for Zeek’s “weird” stream

Zeek has detected that the connection in question does not appear to be standard HTTP. Of note is also the non-standard destination port “8080” under “id.resp_p”, commonly used for HTTP Proxying . We’d usually expect to see port 443 for web traffic, or port 80 in rare legacy cases. Attackers will frequently attempt C2 via ports that have a high probability of being permitted through any network access controls such as a firewall. VirusTotal also confirms that the IP address “*118.110.236.121*” is malicious. Lastly, when we look at the Log Detail for the initial HTTP requests, we see it’s an *HTTP POST* request. The random character URI also seems suspicious:



HTTP POST request to URI including randomized alphanumeric directory name

A safe hunch would be that this is the command and control (C2) traffic we're seeing here. When we filter for our suspected C2 Server "*118.110.236.121*" with the *HTTP POST* method in the Zeek HTTP Stream, we can see the beaconing:



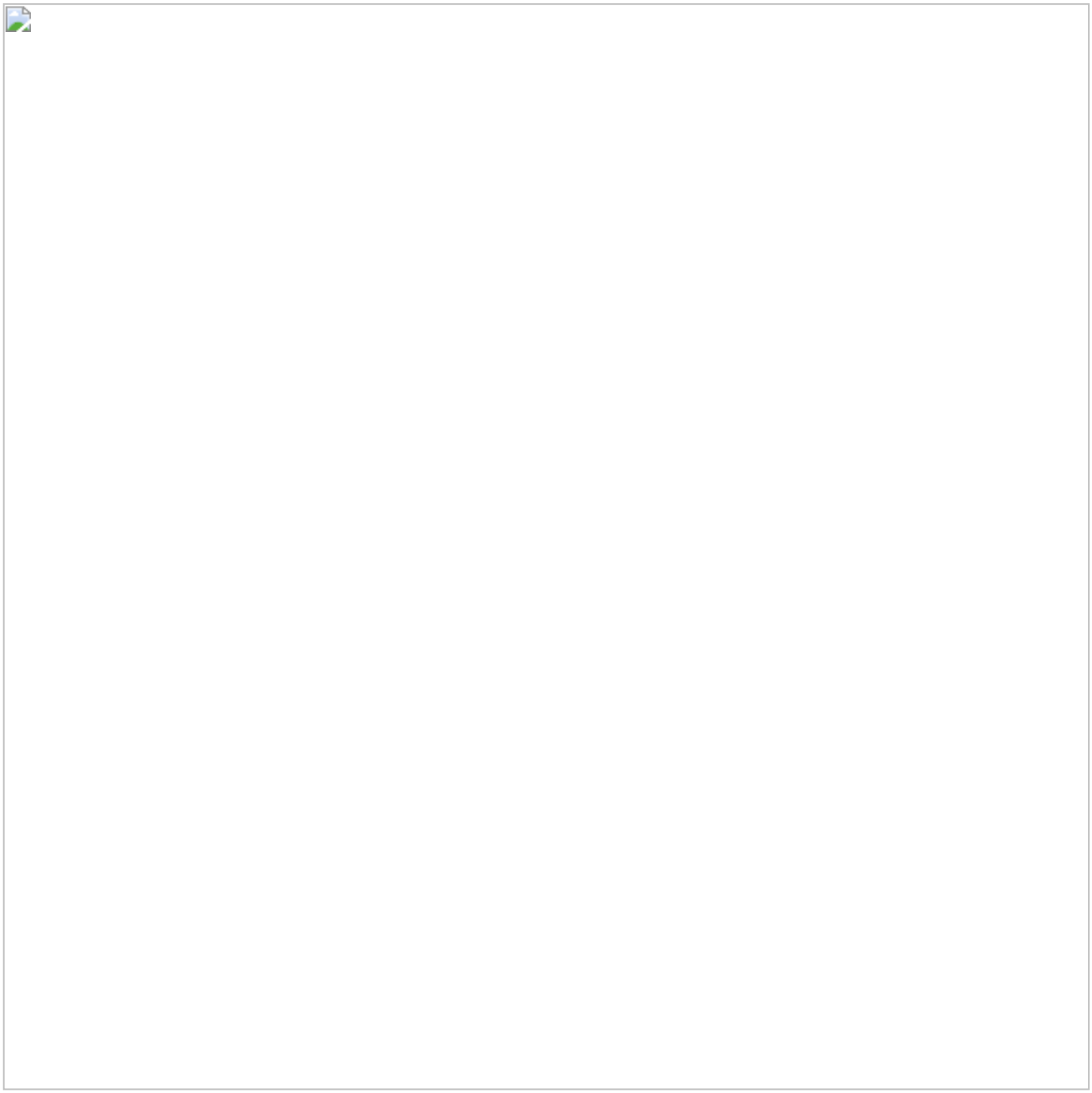
Emotet C2 Traffic

Enumerating Command and Control

> **TIP!** To evade detection and provide redundancy against C2 servers being blocked by inclusion on threat intelligence sources, Malware operators use a network of compromised systems for their command and control infrastructure. The list of C2 servers are constantly updated to try and always stay one step ahead of threat researchers. If you're interested in further details about how Malware C2 works, see this article: https://searchsecurity.techtarget.com/feature/Command-and-control-servers-The-puppet-masters-that-govern-malware

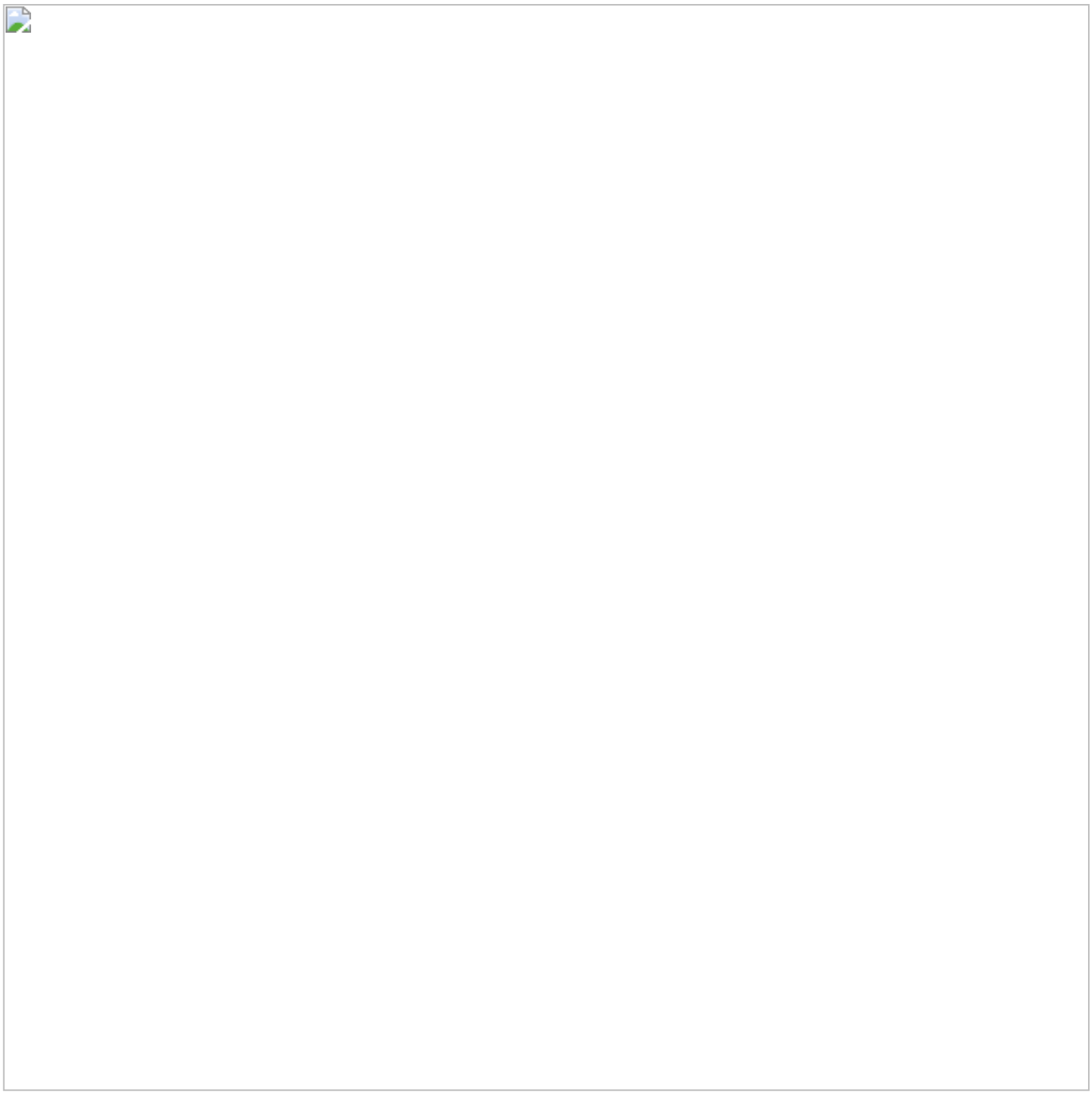
It would be unusual for Emotet to rely on a single C2 instance. One way we can quickly validate this is to search for any additional *HTTP POST* requests Victim X may be sending:

```
id.orig_h=10.9.1.101 method=POST | cut ts, uid, id, method, uri, status_code
```



HTTP POST activity from Patient X

Victim X is concerningly making a number of other successful suspicious *HTTP POST* requests. VirusTotal confirms that all of the involved destination IP addresses are malicious.



VirusTotal confirms our suspicious IP address is malicious

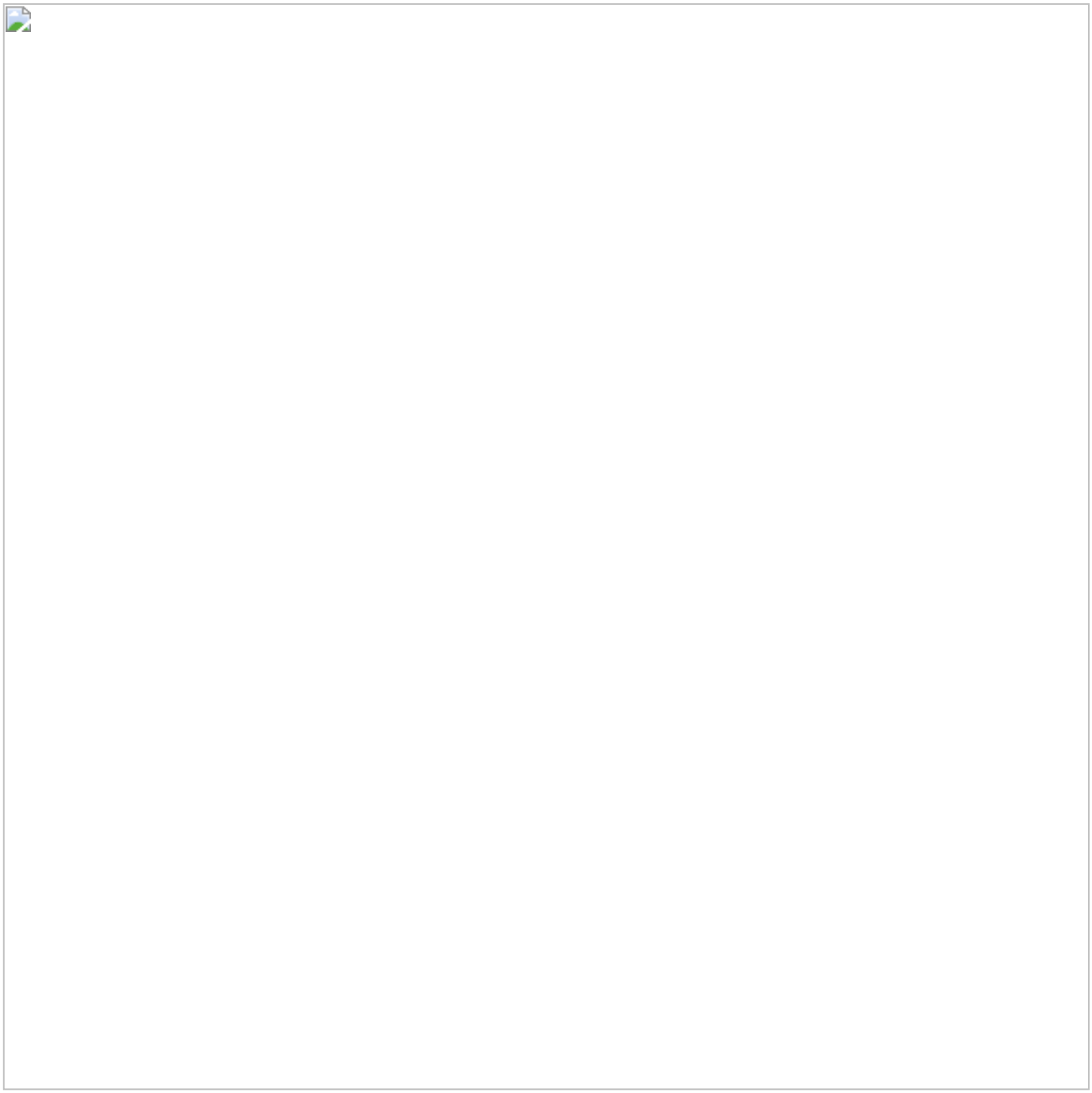
A further search for the malicious IP addresses in our data yields no further results, but of course we can now take our list (*45.230.228.26*, *118.110.236.121*, *195.123.242.119*) and conduct further searches across other detection and search tools, for example a SIEM, as well as adding them to our access control deny list.

Evaluating the spread of Infection

As Emotet is known to propagate via the network, we should also establish whether any other hosts have been infected.

First let's see what sort of traffic our data contains. We can lean on Zeek's streams here again, and generate a list of any contained in the packet capture with associated counts:

```
_path | count() by _path | sort -r
```

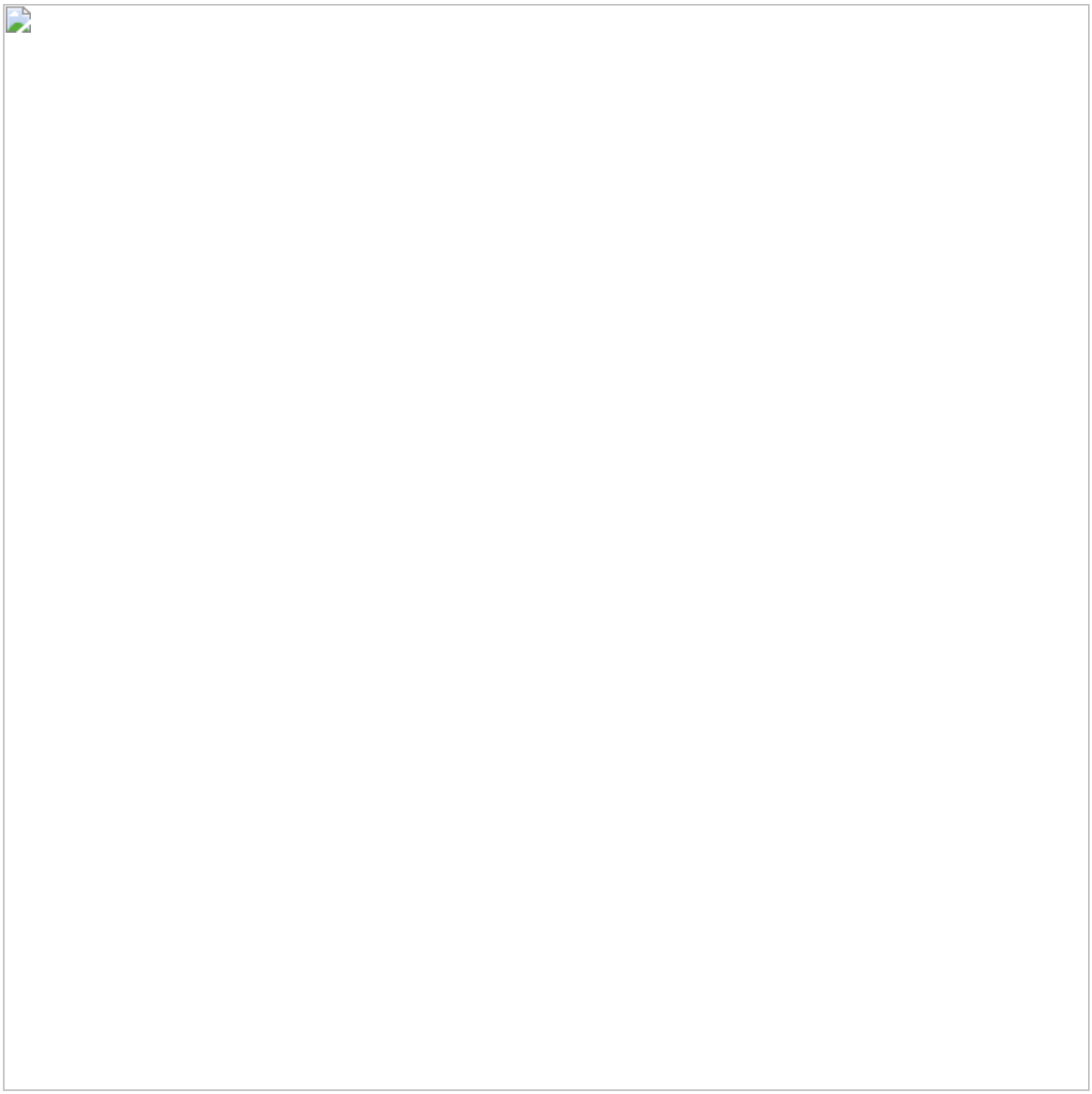


Zeek Streams by count

Interestingly enough, we don't see any activity that would indicate internal reconnaissance or propagation. If present, we'd typically expect to see something like Windows SMB and DCE/RPC activity here. This does not necessarily mean that no further infection occurred — just that we don't have any indicators in our data.

Lastly, just to be sure, we'll enumerate any connections our patient X may have attempted on the internal network. For this, we're going to use ZQL's rich data typing, specifically that there is an IP address data type that supports CIDR filtering:

```
id.orig_h=10.9.1.101 id.resp_h=~10.0.0.0/8 | count() by id.resp_h,_path
```

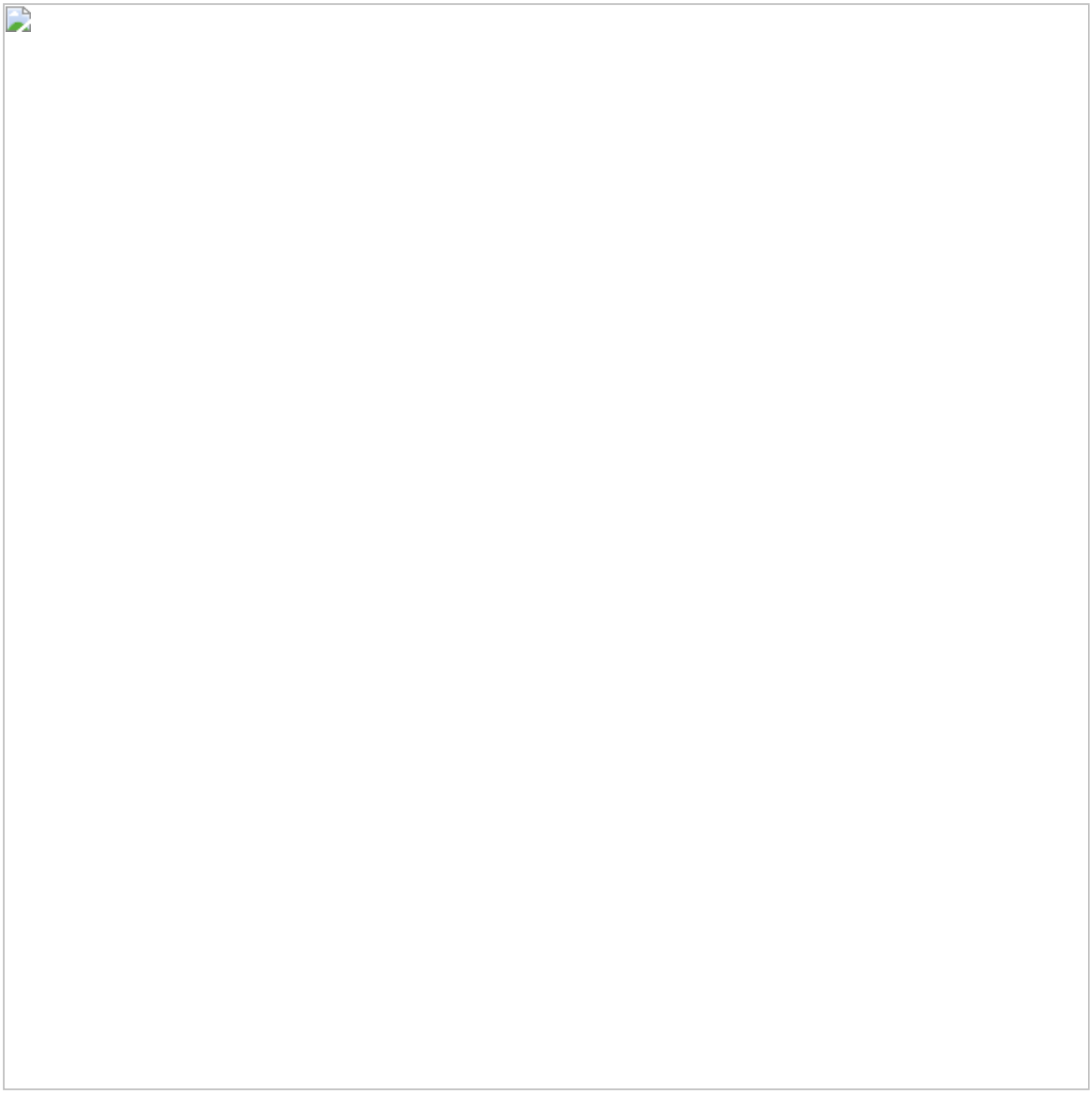


All connections in the 10.0.0.0/8 subnet that patient X communicated with
We see there is some DNS traffic to two other hosts, which we may assume are DNS servers. Nothing in the data indicates further infections.

Putting it all together

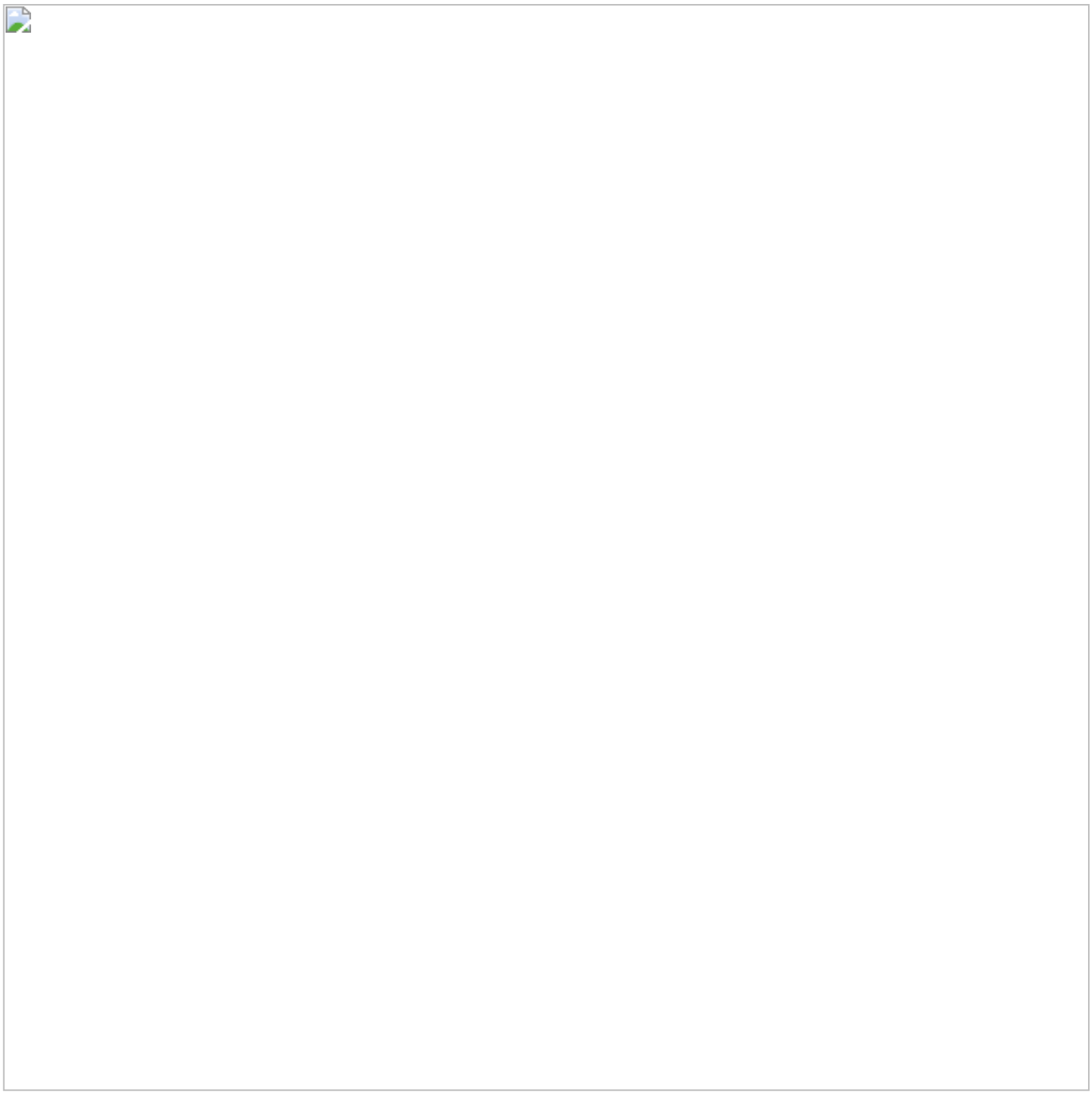
While we didn’t see any propagation events, we did successfully pinpoint the initial infection, identified Patient X and enumerated the C2 servers. This will allow us to conduct further threat hunting, develop detections and create signatures and watch lists for blocking the threat on the network. The diagram below outlines the activity we’ve been investigating:

Emotet Sample observed activity



Emotet Incident outline

Observed Indicators of Compromise



Observed MITRE ATT&CK Techniques

We assume that the malicious file we observed being downloaded from “t-privat.de” was the tail end of an initial phishing attempt. That would entail the following MITRE ATT&CK Techniques:

:002 :

Emotet has been delivered by phishing emails containing links

:001

Emotet has relied upon users clicking on a malicious link delivered through spearphishing.

The following MITRE ATT&CK Tactic was also observed:

:001

Adversaries may communicate using application layer protocols associated with web traffic to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.

Conclusion

We hope you enjoyed our little Emotet safari. There's also a video version of this article, you can find it under https://www.youtube.com/watch?v=CW1rNrd7KYU.

Also, don't forget to check out our last two articles, Five Elegant Brim Queries to Threat Hunt in Zeek Logs and Packet Captures and Investigating Network traffic activity using Brim and Zeek. And watch this space, there's more coming soon!

In the meantime, if you haven't checked out Brim yet, go ahead. It's free and it's Open Source.

https://www.brimsecurity.com/