NurisBot

Category: Threat Hunting
Difficulty: Easy
Date Completed: 2025-04-25

🔍 Summary

This lab involved analyzing Suricata and Zeek logs within Splunk to detect and investigate a malware infection originating from an external attacker. The exercise focused on tracking down the attacker's IP, identifying the compromised host, uncovering malicious file downloads, and correlating indicators across HTTP and DNS traffic.

🔁 Timeline of Events

I will Upload an updated version later today I want to sleep man

🔧 Technical Walkthrough

- **Tools Used**: Splunk, Zeek, Suricata
- **Artifacts Found**:
  - Download of suspicious executable: `/temp/3425[.]exe`
  - Domain associated: `nocomcom[.]com`
  - Multiple malicious files including disguised `kv4[.]txt` file
- **Indicators of Compromise (IOCs)**:
  - Attacker IP: `195[.]88[.]191[.]59`
  - Victim IP: `147[.]32[.]84[.]165`
  - Malicious Domain: `nocomcom[.]com`
  - Malicious File SHA256: `6fbc4d506f4d4e0a64ca09fd826408d3103c1a258c370553583a07a4cb9a6530`

💡 Detection Logic

- **Data Summary Sourcetypes**: *(found from Data Summary menu)*
  Zeek security monitor and Suricata IDS generated all logs.
- **Key Logs**:
  - Zeek: `zeek:files`, `zeek:conn`, `zeek:dns`, `zeek:http`

- Suricata: `event_type=alert`, `event_type=fileinfo`, `event_type=http`

- **Suricata Event Types** (extracted via `index=* sourcetype=suricata | stats count by event_type`):

Event Type	Count	Description
alert	2872	🚨 Detection of attacks or suspicious activity
fileinfo	630	📁 File metadata during transfer
http	1,405	🌍 Web traffic monitoring

- **Important Fields in Alerts**:

Field	Purpose
event_type	Confirmed security events
alert.signature	Attack/malware identification
src_ip, dest_ip	Tracing attacker/victim
flow_id	Correlation between events
app_proto	Application used (HTTP, TLS, etc.)
bytes_in, bytes_out	Potential exfiltration detection

- **Finding our IPs**:

```
index=* sourcetype="suricata"
| fields src_ip, dest_ip
| stats count by src_ip, dest_ip
| sort -count
| head 1000
```

🧐 Lessons Learned

- **What surprised me?**
  No internal IPs made context more confusing; hard to visually map the environment.
- **What would I do differently?**
  Use more cross-correlation between Zeek and Suricata earlier instead of trying HTTP filtering only.
- **Which new tool or concept did I master?**
  - Zeek: how to leverage `zeek:files` for extracting hashes
  - Splunk Query: Advanced `join` to correlate `bytes` between Suricata and Zeek.

🧠 Thought Process for Each Question

Q1: What is the IP address from which the initial unauthorized access originated?

During the investigation of network traffic, unusual patterns of activity were observed in Suricata logs, suggesting potential unauthorized access. One external IP address initiated access attempts and was later seen downloading a suspicious executable file. This activity strongly indicates the origin of the attack.

First, I was thinking of going through HTTP and finding GET requests but then decided to go to Suricata event type `fileinfo` and HTTP method GET, and table the `src_ip + fileinfo name + hash`.
But I had to do this query first:

```
index=* sourcetype=suricata event_type=alert dest_ip=* src_ip=*
| stats count by dest_ip dest_port
| sort -count
| head 50
```

It turned out the dest IP `147.32.84.165` communicated a lot using random dest ports, so it was most likely used as a C2 server.
There was also `147.32.84.171`.

We could see that this IP used two protocols — HTTP and DCERPC — which is dangerous as it can be used for remote control, and if exposed to the internet it would be very dangerous.

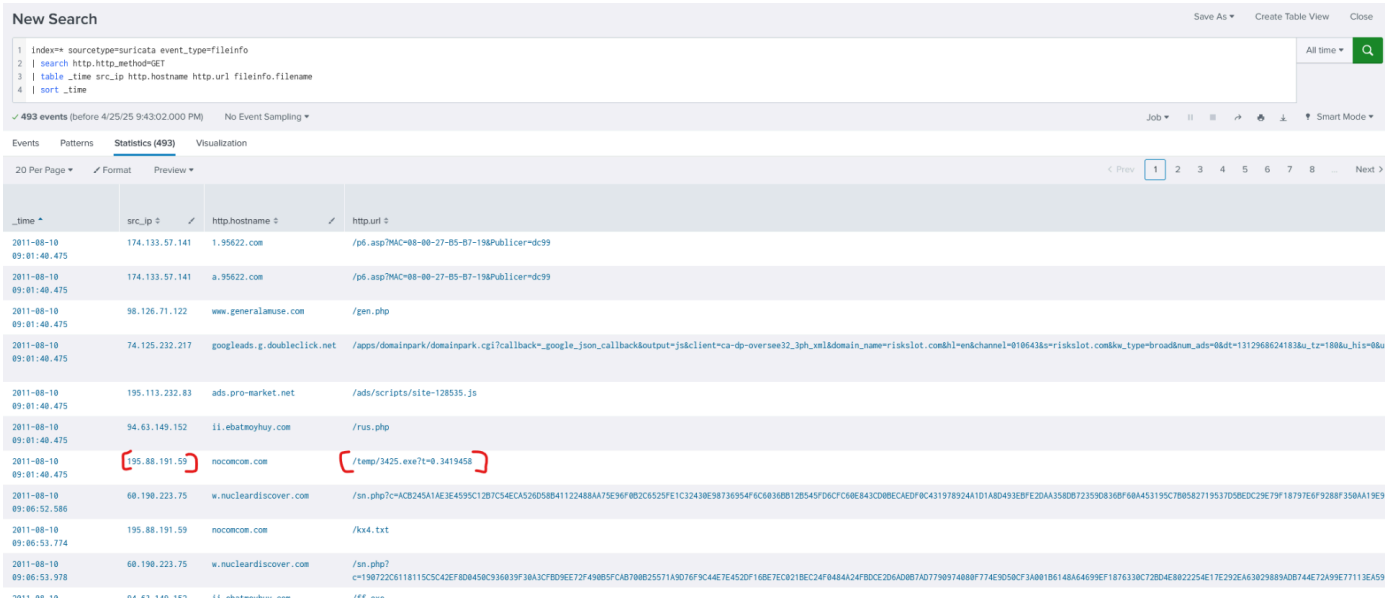
Getting back to the search for the file:

```
index=* sourcetype=suricata event_type=fileinfo
| search http.http_method=GET
| table _time src_ip http.hostname http.url fileinfo.filename
| sort _time
```

Among all the files:

2011-08-10 09:01:40.475	195.88.191.59	nocomcom.com	/temp/3425.exe?t=0.3419458	/temp/3425.exe

This is the most suspicious.

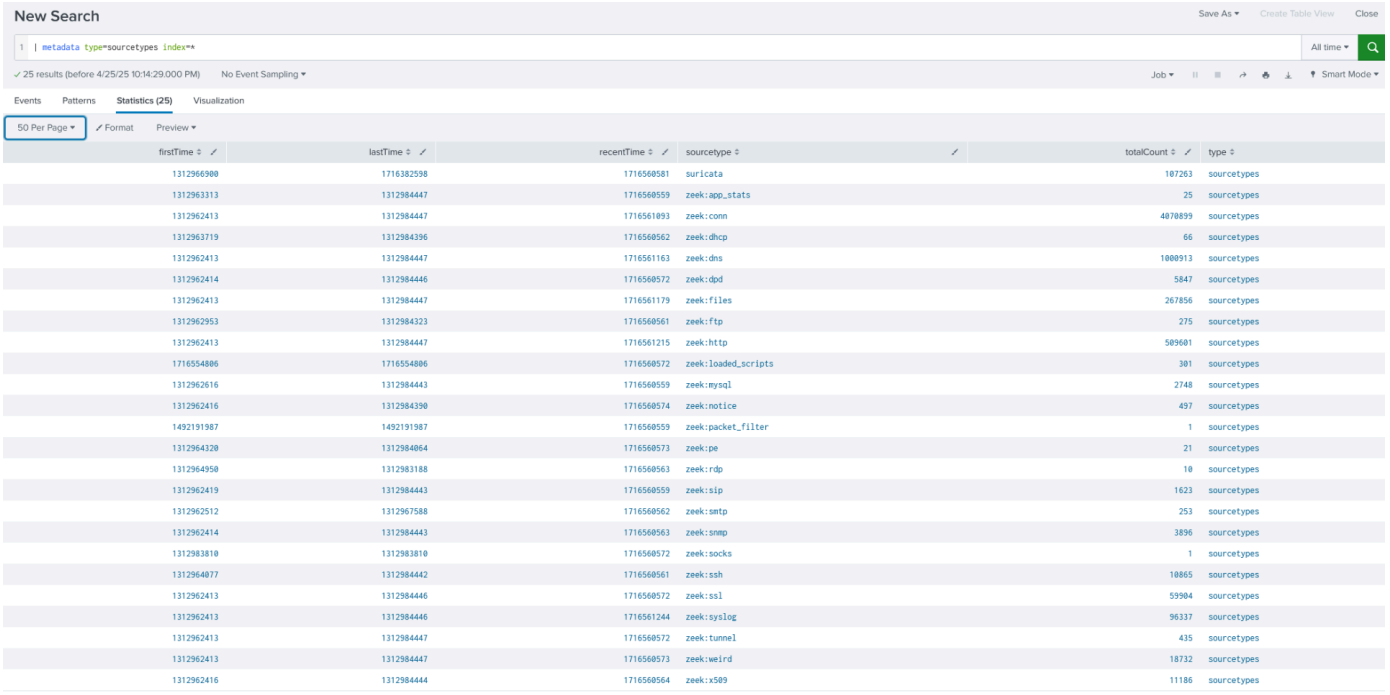


So the IP is most likely `195.88.191.59` and it's correct.

So we now know the attacker IP is `195.88.191.59`.

But tbh the way [SwordFischer] did it was way better:

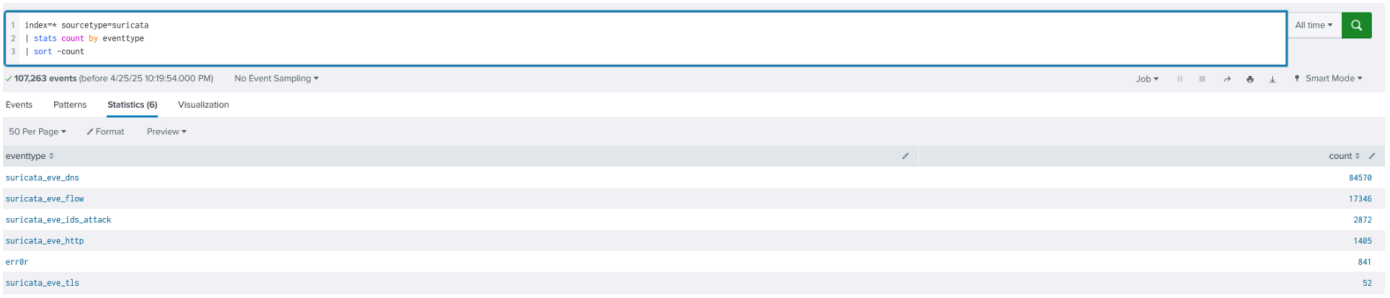
```
| metadata type=sourcetypes index=*
```



This lists all sourcetypes and their counts.

We saw Suricata in there, then:

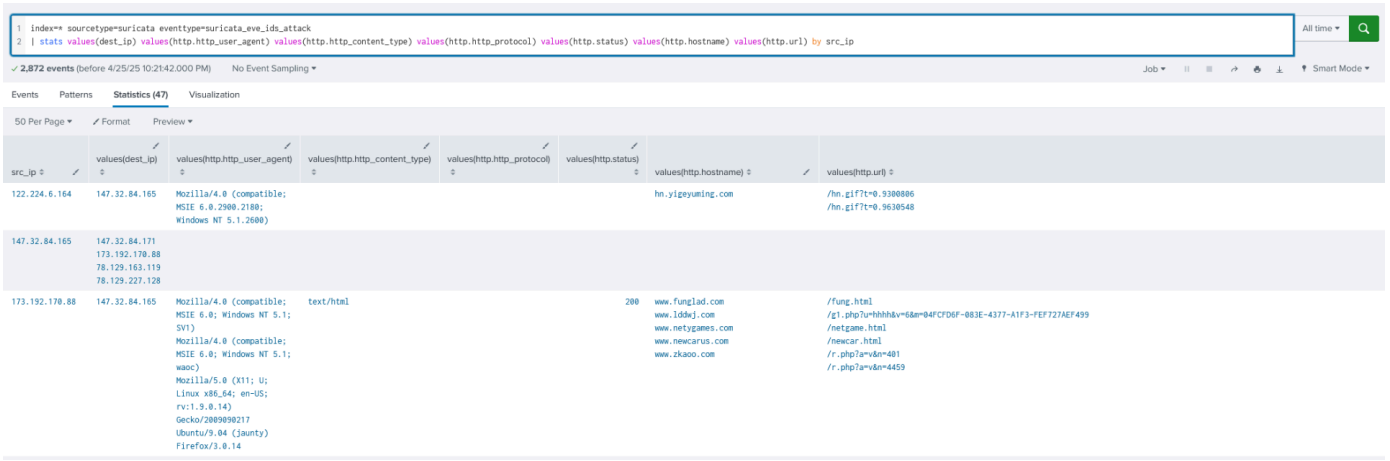
```
index=* sourcetype=suricata
| stats count by eventtype
| sort -count
```



We can see `suricata_eve_ids_attack`.

Then he did:

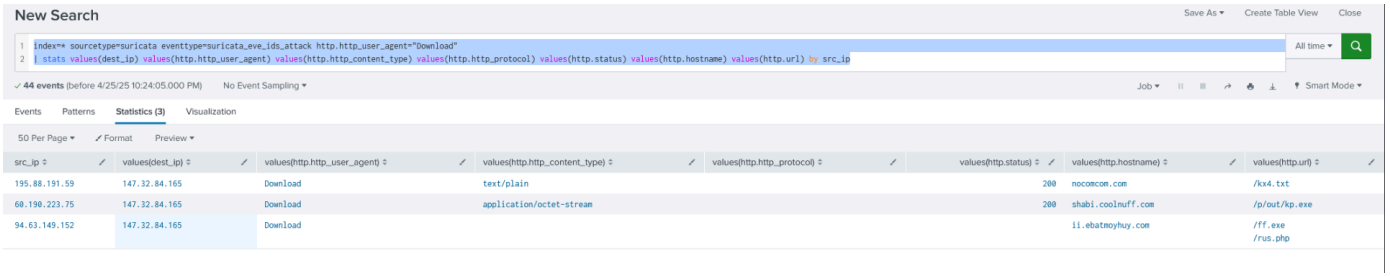
```
index=* sourcetype=suricata eventtype=suricata_eve_ids_attack
| stats values(dest_ip) values(http.http_user_agent)
values(http.http_content_type) values(http.http_protocol)
values(http.status) values(http.hostname) values(http.url) by src_ip
```



Which lists all these infos based on each src_ip.

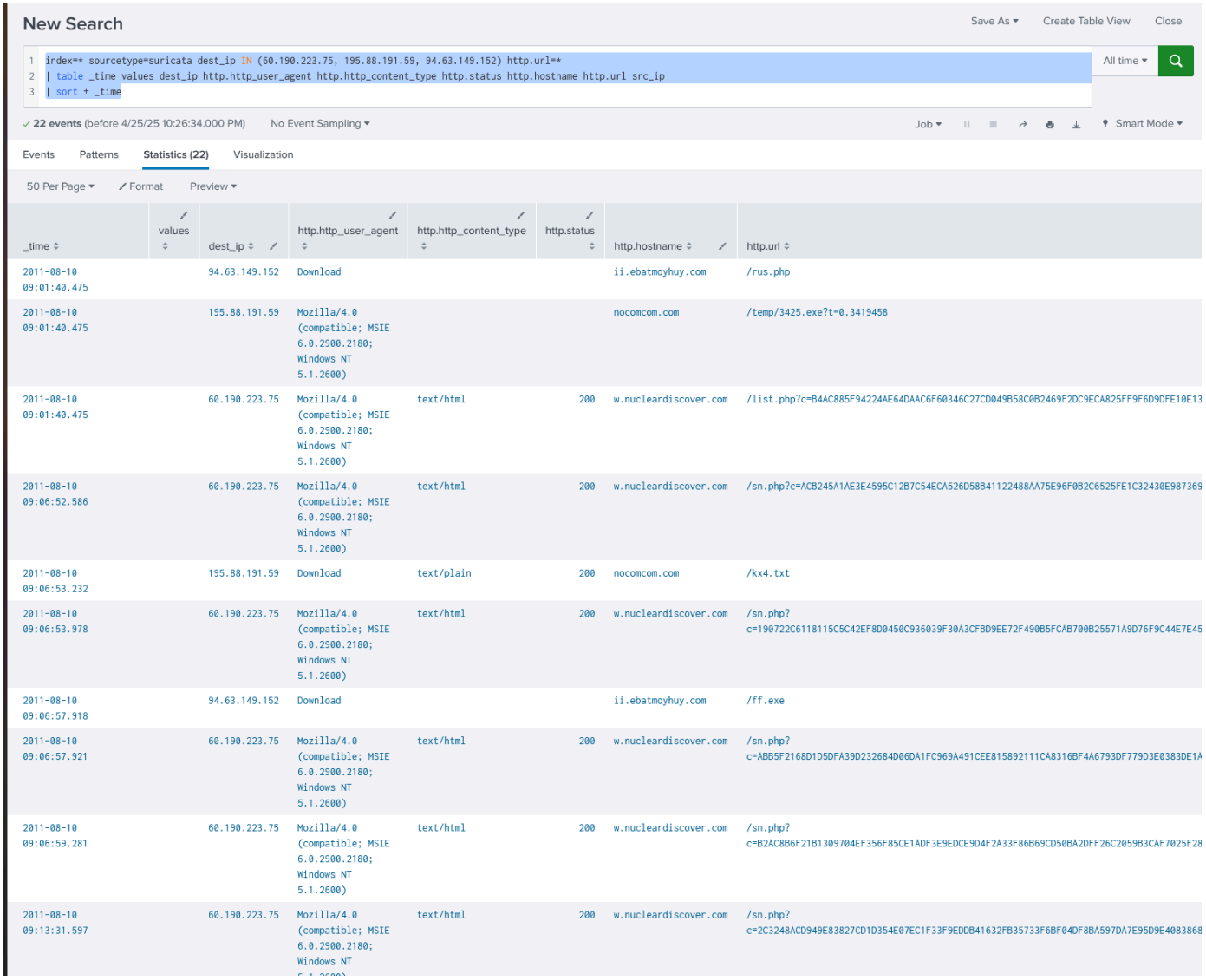
Since we know there is an agent called "Download," we will filter by that:

```
index=* sourcetype=suricata eventtype=suricata_eve_ids_attack
http.http_user_agent="Download"
| stats values(dest_ip) values(http.http_user_agent)
values(http.http_content_type) values(http.http_protocol)
values(http.status) values(http.hostname) values(http.url) by src_ip
```



We have three source IPs that downloaded something.

```
index=* sourcetype=suricata dest_ip IN (60.190.223.75, 195.88.191.59, 94.63.149.152) http.url=*
| table _time dest_ip http.http_user_agent http.http_content_type http.status http.hostname http.url src_ip
| sort +_time
```



New Search

Save As | Create Table View | Close

```
index=* sourcetype=suricata dest_ip IN (60.190.223.75, 195.88.191.59, 94.63.149.152) http.url=*
| table _time values dest_ip http.http_user_agent http.http_content_type http.status http.hostname http.url src_ip
| sort + _time
```

All time

✓ 22 events (before 4/25/25 10:26:34.000 PM) | No Event Sampling | Job | Smart Mode

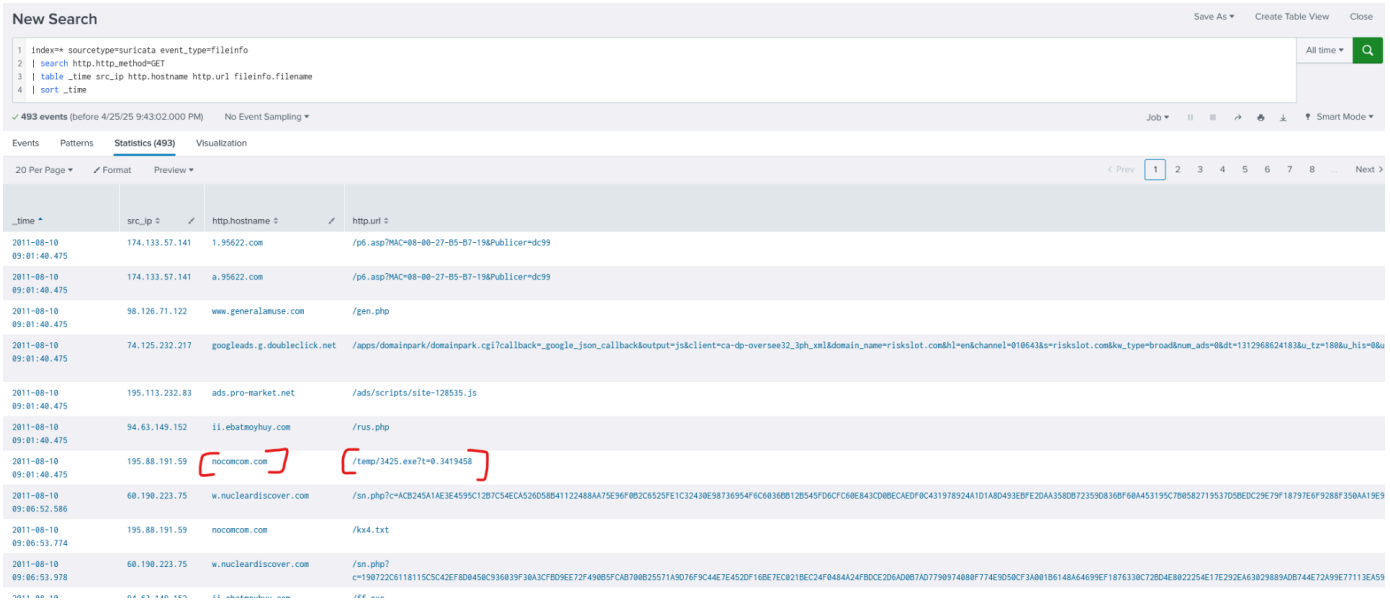
Events | Patterns | Statistics (22) | Visualization

50 Per Page | Format | Preview

_time	values	dest_ip	http.http_user_agent	http.http_content_type	http.status	http.hostname	http.url
2011-08-10 09:01:40.475		94.63.149.152	Download			ii.ebatmoyhuy.com	/rus.php
2011-08-10 09:01:40.475		195.88.191.59	Mozilla/4.0 (compatible; MSIE 6.0.2900.2180; Windows NT 5.1.2600)			nocomcom.com	/temp/3425.exe?t=0.3419458
2011-08-10 09:01:40.475		60.190.223.75	Mozilla/4.0 (compatible; MSIE 6.0.2900.2180; Windows NT 5.1.2600)	text/html	200	w.nucleardiscover.com	/list.php?c=B4AC885F94224AE64DAAC6F60346C27CD049B58C0B2469F2DC9ECA825FF9F6D9DFE10E13
2011-08-10 09:06:52.586		60.190.223.75	Mozilla/4.0 (compatible; MSIE 6.0.2900.2180; Windows NT 5.1.2600)	text/html	200	w.nucleardiscover.com	/sn.php?c=ACB245A1AE3E4595C12B7C54ECA526D58B41122488AA75E96F0B2C6525FE1C32430E98736
2011-08-10 09:06:53.232		195.88.191.59	Download	text/plain	200	nocomcom.com	/kx4.txt
2011-08-10 09:06:53.978		60.190.223.75	Mozilla/4.0 (compatible; MSIE 6.0.2900.2180; Windows NT 5.1.2600)	text/html	200	w.nucleardiscover.com	/sn.php?c=190722C6118115C5C42EF8D0450C936039F30A3CFBD9EE72F490B5FCAB700B25571A9D76F9C44E7E45
2011-08-10 09:06:57.918		94.63.149.152	Download			ii.ebatmoyhuy.com	/ff.exe
2011-08-10 09:06:57.921		60.190.223.75	Mozilla/4.0 (compatible; MSIE 6.0.2900.2180; Windows NT 5.1.2600)	text/html	200	w.nucleardiscover.com	/sn.php?c=ABB5F2168D1D5DFA39D232684D06DA1FC969A491CEE815892111CA8316BF4A6793DF779D3E0383DE1A
2011-08-10 09:06:59.281		60.190.223.75	Mozilla/4.0 (compatible; MSIE 6.0.2900.2180; Windows NT 5.1.2600)	text/html	200	w.nucleardiscover.com	/sn.php?c=B2AC8B6F21B1309704EF356F85CE1ADF3E9EDCE9D4F2A33F86B69CD50BA2DFF26C2059B3CAF7025F28
2011-08-10 09:13:31.597		60.190.223.75	Mozilla/4.0 (compatible; MSIE 6.0.2900.2180; Windows NT	text/html	200	w.nucleardiscover.com	/sn.php?c=2C3248ACD949E83827CD1D354E07EC1F33F9EDDB41632FB35733F6BF04DF8BA597DA7E95D9E4083868

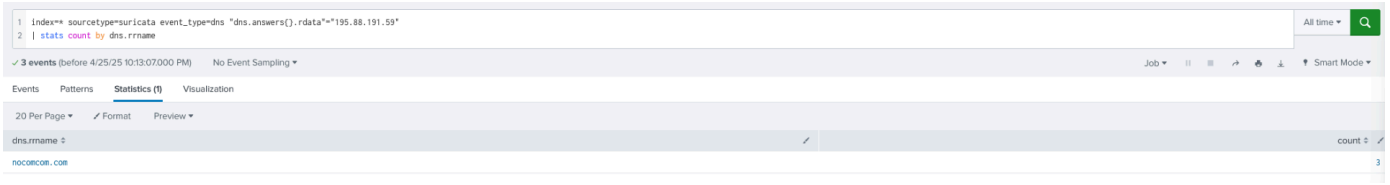
Q2: What is the domain name of the attacker server?

The query that we had above solved our question.



Also, we can do:

```
index=* sourcetype=suricata event_type=dns
"dns.answers{}.rdata"="195.88.191.59"
| stats count by dns.rrname
```



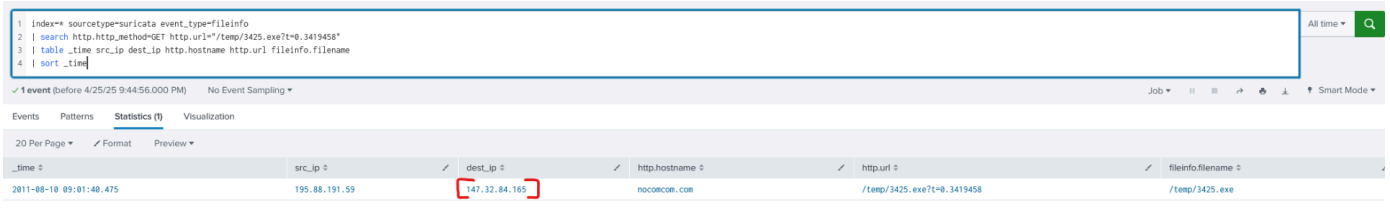
Q3: What is the IP address of the system that was targeted in this breach?

Ok, I made a dumb mistake — by putting `147.32.84.171` instead of `147.32.84.165` as the answer because we already know beforehand that `.165` is the attacked machine.

If I tweaked the query a bit:

```
index=* sourcetype=suricata event_type=fileinfo
| search http.http_method=GET http.url="/temp/3425.exe?t=0.3419458"
| table _time src_ip dest_ip http.hostname http.url fileinfo.filename
| sort _time
```

I will get the dest_ip where that file is sent to.

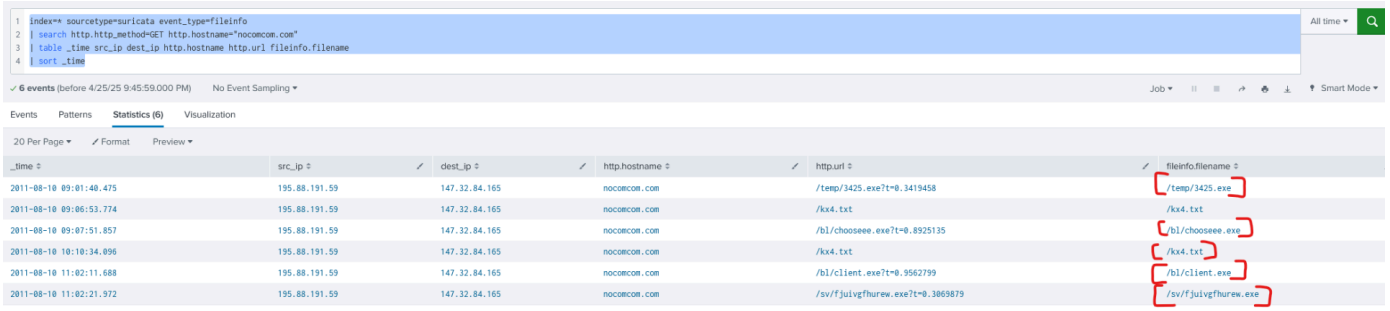


Q4: Identify all the unique files downloaded to the compromised host. How many of these files could potentially be malicious?

For that, I tweaked the query a bit:

```
index=* sourcetype=suricata event_type=fileinfo
| search http.http_method=GET http.hostname="nocomcom.com"
| table _time src_ip dest_ip http.hostname http.url fileinfo.filename
| sort _time
```

Resulted in:

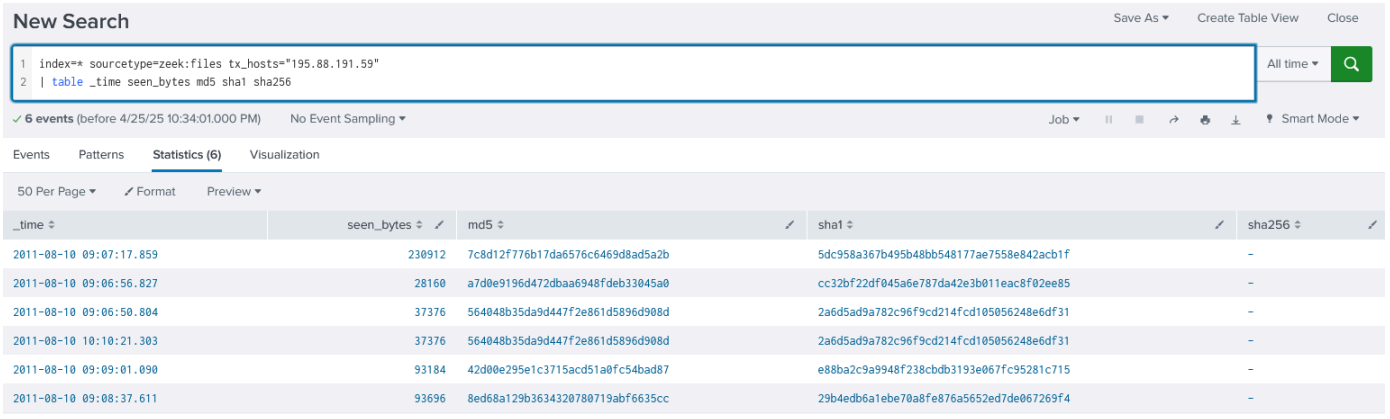


_time	src_ip	dest_ip	http.hostname	http.url	fileinfo.filename
2011-08-10 09:01:40.475	195.88.191.59	147.32.84.165	nocomcom.com	/temp/3425.exe?t=0.3419458	/temp/3425.exe
2011-08-10 09:06:53.774	195.88.191.59	147.32.84.165	nocomcom.com	/kx4.txt	/kx4.txt
2011-08-10 09:07:51.857	195.88.191.59	147.32.84.165	nocomcom.com	/bl/chooseee.exe?t=0.8925135	/bl/chooseee.exe
2011-08-10 10:10:34.096	195.88.191.59	147.32.84.165	nocomcom.com	/kx4.txt	/kx4.txt
2011-08-10 11:02:11.688	195.88.191.59	147.32.84.165	nocomcom.com	/bl/client.exe?t=0.9562799	/bl/client.exe
2011-08-10 11:02:21.972	195.88.191.59	147.32.84.165	nocomcom.com	/sv/fjuivgfhurew.exe?t=0.3069879	/sv/fjuivgfhurew.exe

I got the answer wrong — it was actually **5**.
They included the `.txt` file as potentially malicious.
I thought it would be 4, but turns out it's 5 — we learn from that.

Damn, for this one I missed up hard.
I didn't know Zeek has `zeek:file` which contains hashes.

```
index=* sourcetype=zeek:files tx_hosts="195.88.191.59"
| table _time seen_bytes md5 sha1 sha256
```



_time	seen_bytes	md5	sha1	sha256
2011-08-10 09:07:17.859	230912	7c8d12f776b17da6576c6469d8ad5a2b	5dc958a367b495b48bb548177ae7558e842acb1f	-
2011-08-10 09:06:56.827	28160	a7d0e9196d472dbaa6948fdeb33045a0	cc32bf22df045a6e787da42e3b011eac8f02ee85	-
2011-08-10 09:06:50.804	37376	564048b35da9d447f2e861d5896d908d	2a6d5ad9a782c96f9cd214fcd105056248e6df31	-
2011-08-10 10:10:21.303	37376	564048b35da9d447f2e861d5896d908d	2a6d5ad9a782c96f9cd214fcd105056248e6df31	-
2011-08-10 09:09:01.090	93184	42d00e295e1c3715acd51a0fc54bad87	e88ba2c9a9948f238cbdb3193e067fc95281c715	-
2011-08-10 09:08:37.611	93696	8ed68a129b3634320780719abf6635cc	29b4edb6a1ebe70a8fe876a5652ed7de067269f4	-

---

Hashes found:

- Hash 1: `42d00e295e1c3715acd51a0fc54bad87`
- Hash 2: `8ed68a129b3634320780719abf6635cc`

- Hash 3: `7c8d12f776b17da6576c6469d8ad5a2b`
- Hash 4: `a7d0e9196d472dbaa6948fdeb33045a0`
- Hash 5: `b5f3729e5418905ad2b21ce186b1c01d`
- Hash 6: `564048b35da9d447f2e861d5896d908d`

Q5: What is the SHA256 hash of the malicious file disguised as a `.txt` file?

So I'm out of luck — really, Suricata `fileinfo` doesn't have the hash for `kx4.txt`.
I have to find another way in my environment to get hashes and I don't want to throw random queries.

Tried:

```
index=* kx4.txt
```

and

```
index=* sourcetype="suricata" url="/kx4.txt"
```

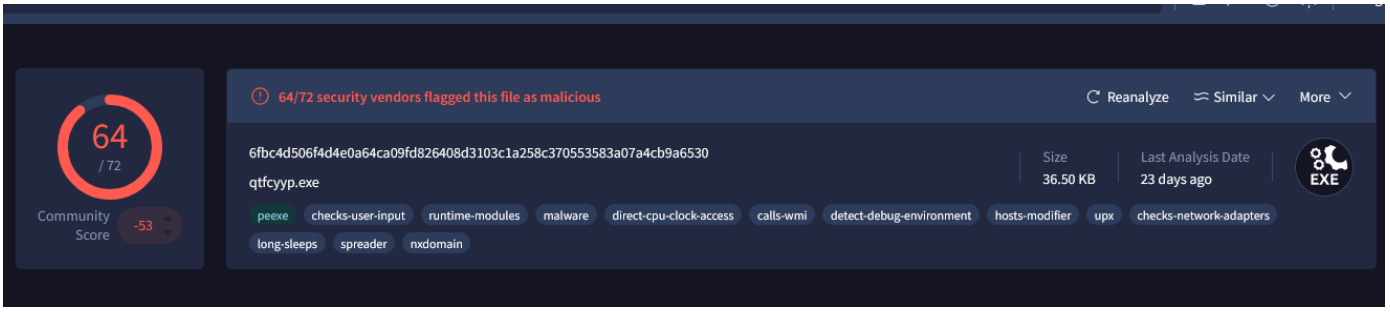
11 results but no hash.

So I ended up going to this report:

> https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-54/botnet-capture-20110815-fast-flux-2.html

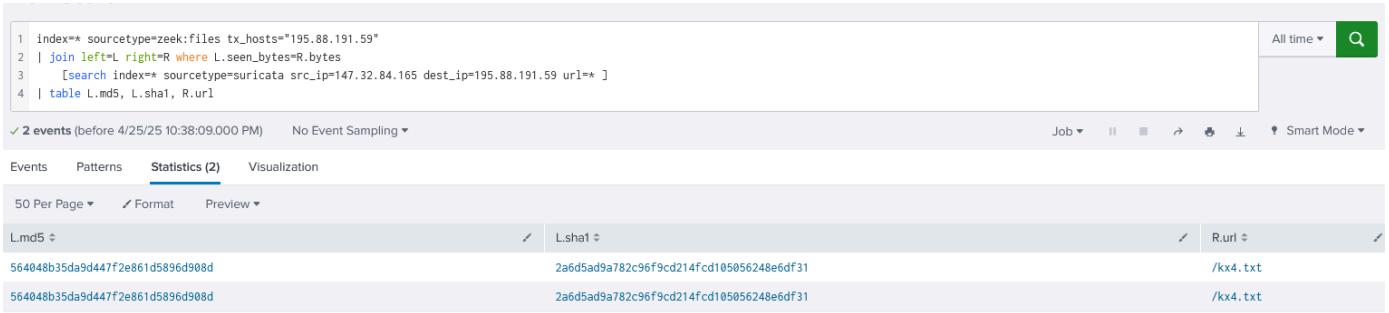
and found the hash.

6fbc4d506f4d4e0a64ca09fd826408d3103c1a258c370553583a07a4cb9a6530	



Also learned this amazing query:

```
index=* sourcetype=zeek:files tx_hosts="195.88.191.59"
| join left=L right=R where L.seen_bytes=R.bytes
    [search index=* sourcetype=suricata src_ip=147.32.84.165
dest_ip=195.88.191.59 url=*]
| table L.md5, L.sha1, R.url
```



This query matches the results from suricata bytes field and from zeek seen_bytes to get one table that have url and hash in it