LAPORAN PRAKTIKUM KEAMANAN INFORMASI 1

Pertemuan 5

IP & *Enterprise Services Vulnerability*

DISUSUN OLEH

Nama	: Sofiyanatul Munawaroh
NIM	: 21/474781/SV/19035
Hari, Tanggal	: Selasa, 14 Maret 2023
Kelas	: RI4AA

LABORATORIUM PERANGKAT KERAS DAN LUNAK

PROGRAM SARJANA TERAPAN (DIV) TEKNOLOGI

REKAYASA INTERNET

DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS GADJAH MADA

2023

Praktikum Keamanan Informasi 1

Pertemuan 5 – IP & *Enterprise Services Vulnerability*

I. Tujuan

- Menginvestigasi SQL *Injection Attack.*
- Menganalisis *Pre-Captured Logs* dan *Traffic Captures*.
- Menginvestigasi DNS *Data Exfiltration.*

II. Latar Belakang

Internet protocol merupakan sebuah standar atau aturan yang digunakan dalam jaringan untuk mengatur serta mengizinkan terjadinya hubungan antar komputer dan perpindahan data. Komputer-komputer tersebut menjadi dapat saling berkomunikasi antara yang satu dengan yang lainnya dan saling bertukar informasi karena adanya *Internet Protocol*. Adapun jenis-jenis IP yaitu DNS, *Proxy*, IP *address* dan jenis *protocol* lainnya.

Protocol jaringan adalah aturan yang berada di dalam komputer yang mana harus ditaati oleh si pengirim dan penerima. *Protocol* sendiri mempunyai ragam fungsi di dalam jaringan komputer.

Namun, tidak semua *protocol* memiliki fungsi yang sama. Beberapa di antaranya memiliki fungsi yang sama tapi berada di tingkatan yang berbeda. Untuk membangun sistem komunikasi yang utuh sejumlah protocol terlebih dahulu harus bergabung dengan *protocol* lainnya.

Secara umum fungsi *internet protocol* yakni untuk menghubungkan antara pengirim dengan penerima dalam berkomunikasi serta bertukar informasi agar bisa berjalan dengan akurat dan lancar.

IP juga memiliki beberapa kelemahan atau celah keamanan yang dapat dieksploitasi oleh penyerang. *Enterprise services vulnerability* mengacu pada kelemahan atau celah keamanan dalam layanan-layanan yang digunakan dalam lingkungan perusahaan atau bisnis. Layanan-layanan ini mencakup perangkat lunak, sistem operasi, infrastruktur jaringan, dan aplikasi yang digunakan dalam operasi perusahaan. Kerentanan dalam layanan-layanan ini dapat menyebabkan risiko keamanan yang signifikan, seperti kebocoran data, akses yang tidak sah, atau serangan terhadap sistem.

III. Alat dan Bahan

- *Wireshark*
- *Security Onion Virtual Machine*
- *CyberOps Workstation Virtual Machine*
- Laptop/PC
- Koneksi Internet

IV. Instruksi Kerja

A. Menganalisis *Log* yang Ditangkap Sebelumnya dan Pengambilan Lalu Lintas

1. Ubah direktori ke *folder* ‘lab.support.files/pcaps’, dan dapatkan daftar *file* menggunakan perintah ls –l.

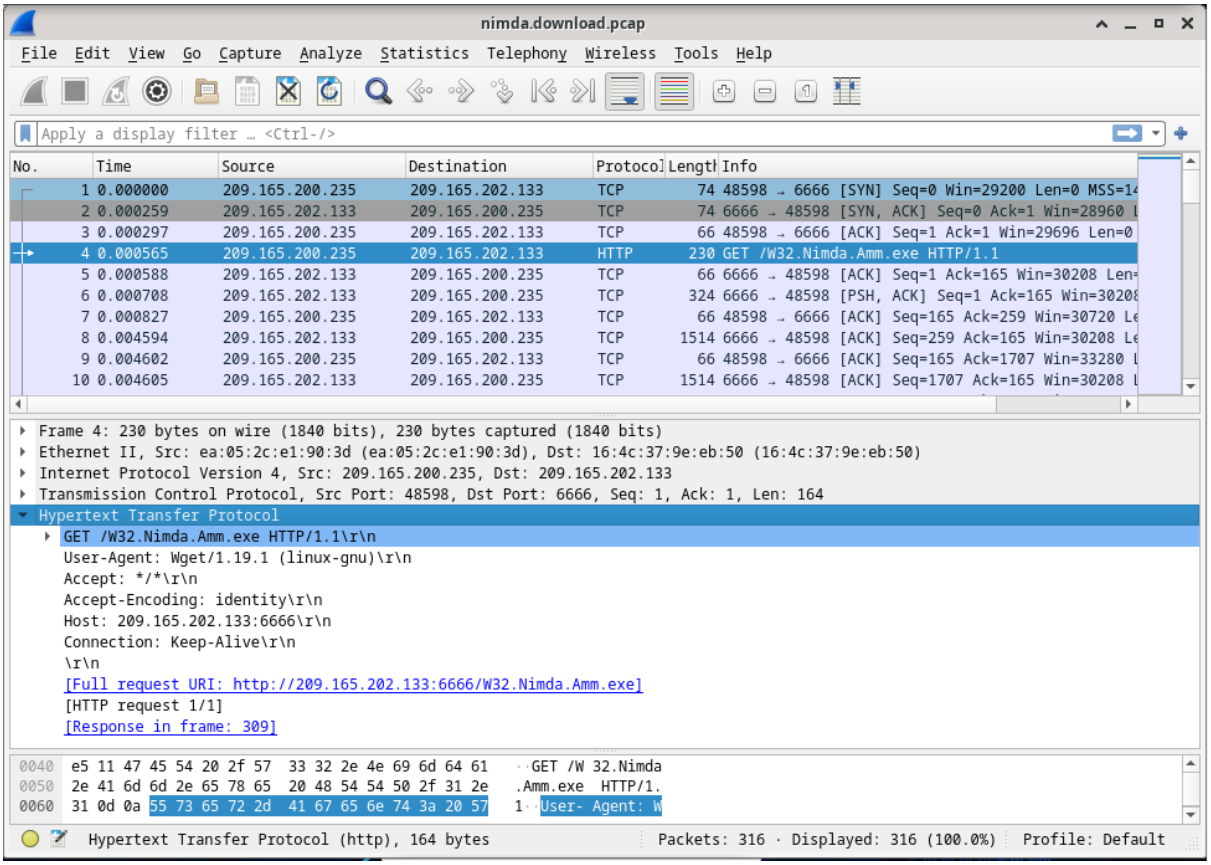
[analyst@secOps ~]$ cd lab.support.files/pcaps
[analyst@secOps pcaps]$ ls -l

```
[analyst@secOps ~]$ cd lab.support.files/pcaps
[analyst@secOps pcaps]$ ls -l
total 4028
-rw-r--r-- 1 analyst analyst  371462 Mar 21  2018 nimda.download.pcap
-rw-r--r-- 1 analyst analyst 3750153 Mar 21  2018 wannacry_download_pcap.pcap
```

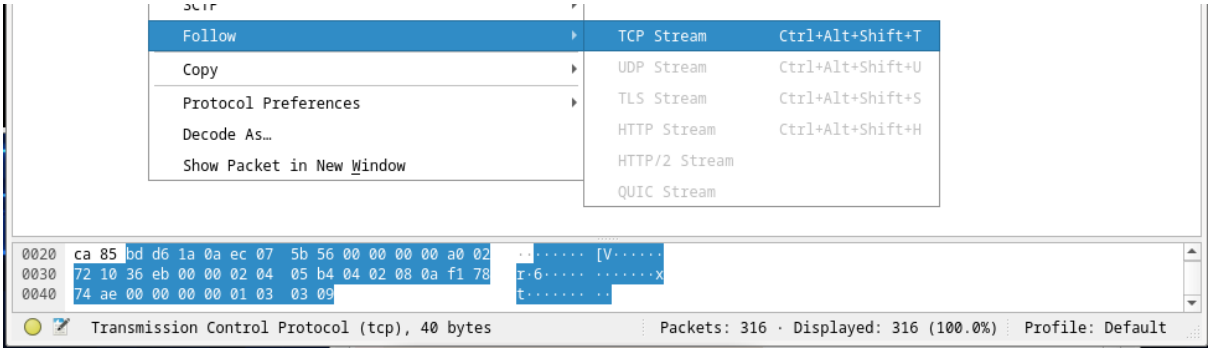
2. Keluarkan perintah di bawah ini untuk membuka *file* 'nimda.download.pcap' di Wireshark.
   [analyst@secOps pcaps]$ wireshark nimda.download.pcap &

```
[analyst@secOps pcaps]$ wireshark nimda.download.pcap &
[1] 541
```

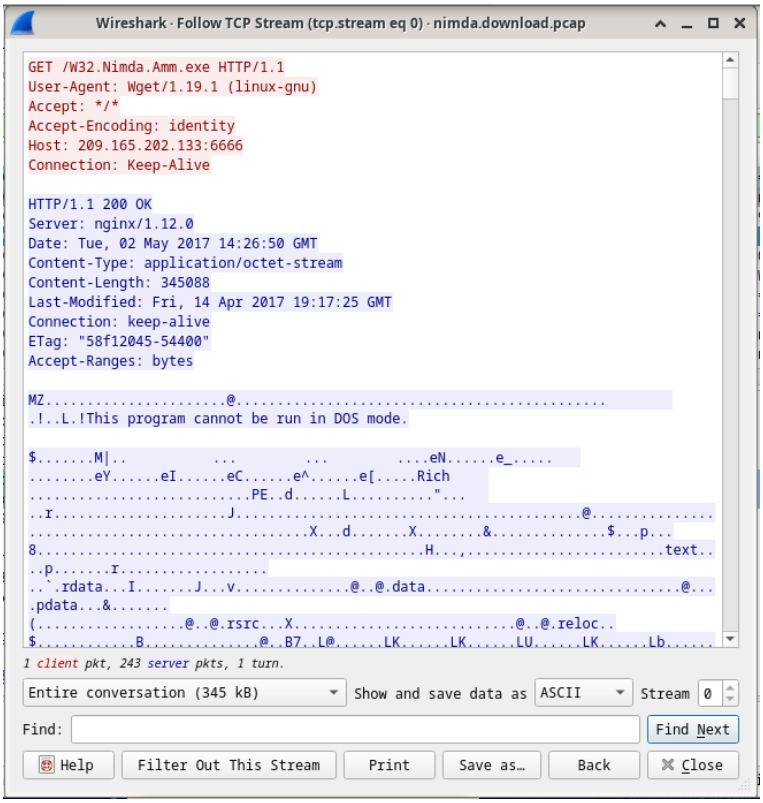
3. *File* 'nimda.download.pcap' berisi pengambilan paket yang terkait dengan unduhan *malware*. Pcap berisi semua paket yang dikirim dan diterima saat tcpdump sedang berjalan. Pilih paket keempat dalam tangkapan dan perluas Protokol *Transfer Hypertext* untuk ditampilkan seperti yang ditunjukkan di bawah ini.



4. Karena HTTP berjalan di atas TCP, dimungkinkan untuk menggunakan fitur *Follow* TCP *Stream* Wireshark untuk membangun kembali transaksi TCP. Pilih paket TCP pertama yang di-*capture*, paket SYN. Klik kanan dan pilih Ikuti > TCP *Stream*.

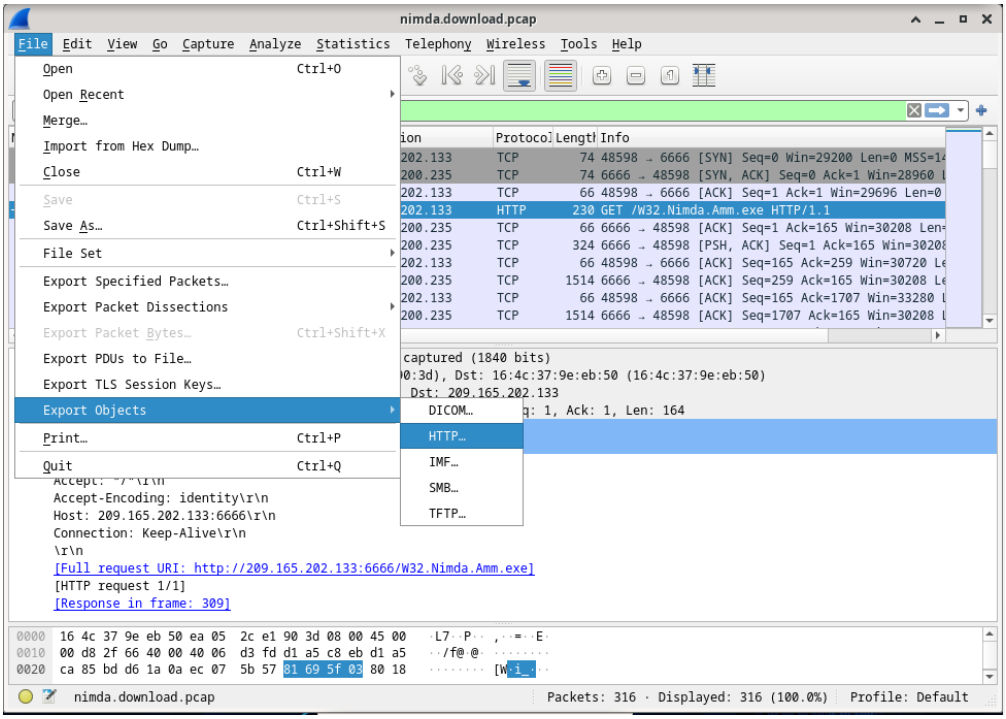


5. Wireshark menampilkan jendela lain yang berisi detail untuk seluruh aliran TCP yang dipilih.

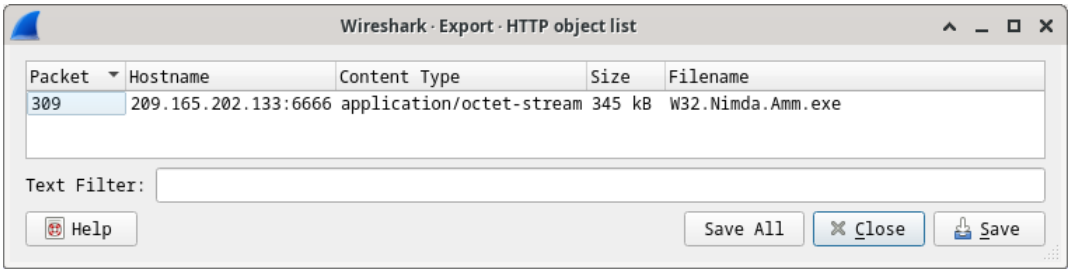


B. *Extract Files* yang Diunduh dari PCAP

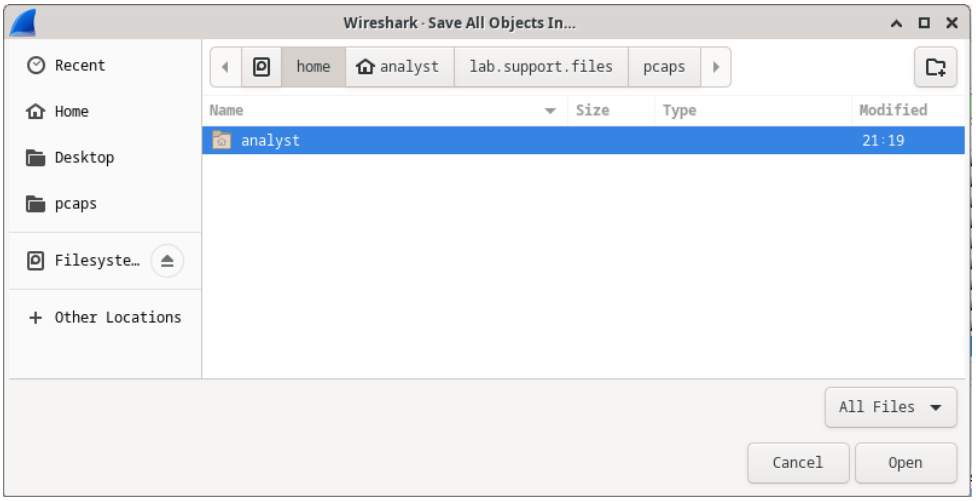
6. Dalam paket keempat dalam *file* nimda.download.pcap, perhatikan bahwa permintaan HTTP GET dihasilkan dari 209.165.200.235 menjadi 209.165.202.133. Kolom Info juga menunjukkan bahwa ini sebenarnya adalah permintaan GET untuk *file* tersebut.
7. Dengan paket permintaan GET yang dipilih, navigasikan ke *File* > *Export Objects* > HTTP, dari menu Wireshark.



8. Wireshark akan menampilkan semua objek HTTP yang ada dalam aliran TCP yang berisi permintaan GET. Dalam hal ini, hanya *file* W32.Nimda.Amm.exe yang ada dalam pengambilan. Ini akan memakan waktu beberapa detik sebelum *file* ditampilkan.



9. Di jendela daftar objek HTTP, pilih *file* W32.Nimda.Amm.exe dan klik '*Save As'* di bagian bawah layar.
10. Klik panah kiri hingga melihat tombol Beranda. Klik Beranda lalu klik *folder* analis (bukan *tab* analis). Simpan *file* di sana.



11. Kembali ke jendela terminal Anda dan pastikan *file* telah disimpan. Ubah direktori ke *folder* /home/analyst dan daftarkan *file* di folder tersebut menggunakan perintah ls -l.

```
[analyst@secOps pcaps]$ cd /home/analyst
[analyst@secOps ~]$ ls -l
total 18776
drwxr-xr-x 2 analyst analyst     4096 May 20  2020 Desktop
drwxr-xr-x 3 analyst analyst     4096 Apr  2  2020 Downloads
-rw-r--r-- 1 root    root     6261934 Feb 20 21:05 httpdump.pcap
-rw-r--r-- 1 root    root    12595200 Feb 20 21:33 httpsdump.pcap
drwxr-xr-x 9 analyst analyst     4096 Jul 15  2020 lab.support.files
drwxr-xr-x 2 analyst analyst     4096 Feb 15 21:07 lalala
drwxr-xr-x 2 analyst analyst     4096 Mar 21  2018 second_drive
-rw-r--r-- 1 analyst analyst   345088 Mar 13 22:11 W32.Nimda.Amm.exe
[analyst@secOps ~]$
```

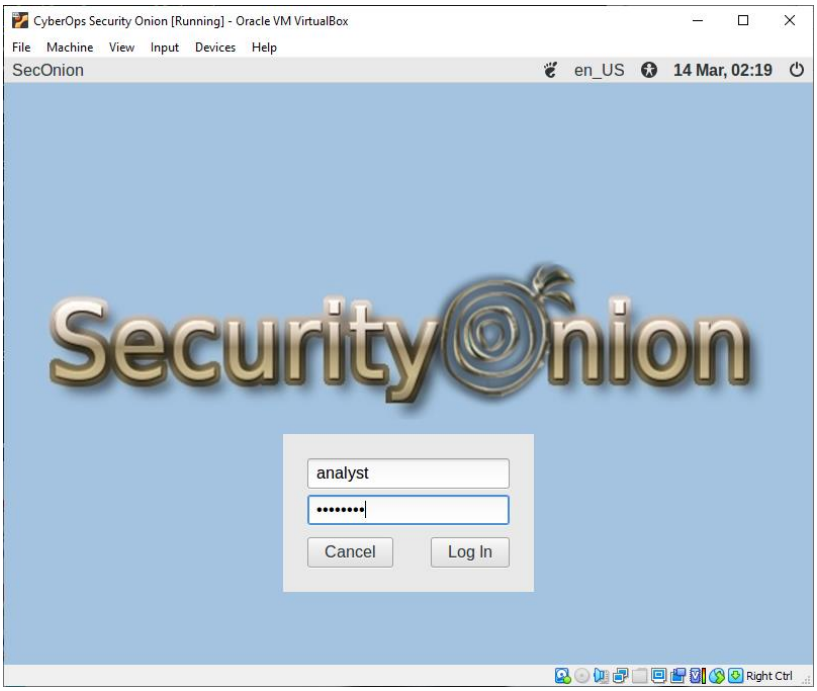
12. Perintah *file* memberikan informasi tentang jenis *file*. Gunakan perintah *file* untuk mempelajari lebih lanjut tentang *malware*, seperti yang ditunjukkan di bawah ini: [analyst@secOps ~]$ file W32.Nimda.Amm.exe

```
[analyst@secOps ~]$ file W32.Nimda.Amm.exe
W32.Nimda.Amm.exe: PE32+ executable (console) x86-64, for MS Windows
```

C. Persiapan *Log File* pada *Security Onion Virtual Machine*

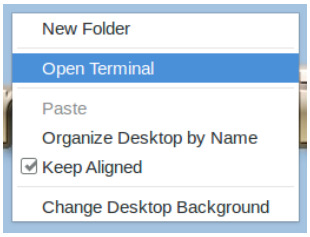
a. *Security Onion VM*

Luncurkan *Security Onion VM* dengan *username*: **analyst,** *password*: **cybercops.**

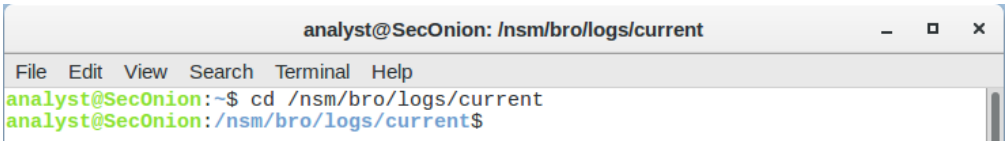


b. *Zeek Logs* pada *Security Onion*

1. Buka jendela terminal dengan klik kanan *Desktop*, pilih *Open Terminal.*



2. *Log Zeek* disimpan di /nsm/bro/logs/. *File log* saat ini dapat ditemukan di bawah direktori saat ini. Dari jendela terminal, ubah direktori menggunakan perintah berikut.
   analyst@SecOnion:~$ cd /nsm/bro/logs/current
   analyst@SecOnion:/nsm/logs/current$



3. Gunakan perintah ls -l untuk melihat *file log* yang dihasilkan oleh Zeek.

```
analyst@SecOnion:/nsm/bro/logs/current$ ls -l
total 0
analyst@SecOnion:/nsm/bro/logs/current$
```

c. *Snort Logs*

1. *Log snort* dapat ditemukan di /nsm/sensor_data/. Ubah direktori sebagai berikut.
   analyst@SecOnion:/nsm/bro/logs/current$ cd /nsm/sensor_data
   analyst@SecOnion:/nsm/sensor_data$

```
analyst@SecOnion:/nsm/bro/logs/current$ cd /nsm/sensor_data
analyst@SecOnion:/nsm/sensor_data$
```

2. Gunakan perintah ls -l untuk melihat semua *file log* yang dihasilkan oleh *Snort.*

```
analyst@SecOnion:/nsm/sensor_data$ ls -l
total 12
drwxrwxr-x 7 sguil sguil 4096 Jun 19  2020 seconion-eth0
drwxrwxr-x 5 sguil sguil 4096 Jun 19  2020 seconion-eth1
drwxrwxr-x 7 sguil sguil 4096 Jun 19  2020 seconion-import
analyst@SecOnion:/nsm/sensor_data$
```

3. Perhatikan bahwa *Security Onion* memisahkan *file* berdasarkan antarmuka. Karena *image Security Onion* VM memiliki dua antarmuka yang dikonfigurasi sebagai sensor dan *folder* khusus untuk data yang diimpor, tiga direktori disimpan. Gunakan perintah ls –l seconion-eth0 untuk melihat *file* yang dihasilkan oleh antarmuka eth0.

```
analyst@SecOnion:/nsm/sensor_data$ ls -l seconion-eth0
total 28
drwxrwxr-x 2 sguil sguil 4096 Jun 19  2020 argus
drwxrwxr-x 3 sguil sguil 4096 Jun 19  2020 dailylogs
drwxrwxr-x 2 sguil sguil 4096 Jun 19  2020 portscans
drwxrwxr-x 2 sguil sguil 4096 Jun 19  2020 sancp
drwxr-xr-x 2 sguil sguil 4096 Jun 19  2020 snort-1
-rw-r--r-- 1 sguil sguil 5594 Jun 19  2020 snort-1.stats
-rw-r--r-- 1 root  root     0 Jun 19  2020 snort.stats
analyst@SecOnion:/nsm/sensor_data$
```

d. *Various Logs*

1. Sementara direktori /nsm/ menyimpan beberapa *file log, file log* yang lebih spesifik dapat ditemukan di bawah /var/log/nsm/. Ubah direktori dan gunakan perintah ls untuk melihat semua *file log* di direktori.

```
analyst@SecOnion:/nsm/sensor_data$ cd /var/log/nsm/
analyst@SecOnion:/var/log/nsm$ ls
eth0-packets.log        sensor-newday-argus.log
netsniff-sync.log       sensor-newday-http-agent.log
ossec_agent.log         sensor-newday-pcap.log
seconion-eth0           so-elastic-configure-kibana-dashboards.log
seconion-import         so-elasticsearch-pipelines.log
securityonion           sosetup.log
sensor-clean.log        so-zeek-cron.log
sensor-clean.log.1.gz   squert-ip2c-5min.log
sensor-clean.log.2.gz   squert-ip2c.log
sensor-clean.log.3.gz   squert_update.log
sensor-clean.log.4.gz   watchdog.log
sensor-clean.log.5.gz   watchdog.log.1.gz
sensor-clean.log.6.gz   watchdog.log.2.gz
sensor-clean.log.7.gz
analyst@SecOnion:/var/log/nsm$
```

2. *Log* ELK dapat ditemukan di direktori /var/log. Ubah direktori dan gunakan perintah ls untuk membuat daftar *file* dan direktori.

```
analyst@SecOnion:/var/log/nsm$ cd ..
analyst@SecOnion:/var/log$ ls
alternatives.log        daemon.log        fsck              salt
alternatives.log.1      daemon.log.1      gpu-manager.log   samba
alternatives.log.2.gz   daemon.log.2.gz   installer         sguild
alternatives.log.3.gz   daemon.log.3.gz   kern.log          so-boot.log
alternatives.log.4.gz   daemon.log.4.gz   kern.log.1        syslog
alternatives.log.5.gz   debug             kern.log.2.gz     syslog.1
apache2                 debug.1           kibana            syslog.2.gz
apt                     debug.2.gz        lastlog           syslog.3.gz
auth.log                debug.3.gz        lightdm           syslog.4.gz
auth.log.1              debug.4.gz        logstash          syslog.5.gz
auth.log.2.gz           dmesg             lpr.log           syslog.6.gz
auth.log.3.gz           domain_stats      mail.err          syslog.7.gz
auth.log.4.gz           dpkg.log          mail.info         unattended-upgrades
boot                    dpkg.log.1        mail.log          user.log
boot.log                elastalert        mail.warn         user.log.1
bootstrap.log           elasticsearch     messages          user.log.2.gz
btmp                    error             messages.1        user.log.3.gz
btmp.1                  error.1           messages.2.gz     user.log.4.gz
cron.log                error.2.gz        messages.3.gz     wtmp
cron.log.1              error.3.gz        messages.4.gz     wtmp.1
cron.log.2.gz           error.4.gz        mysql             Xorg.0.log
cron.log.3.gz           faillog           nsm               Xorg.0.log.old
cron.log.4.gz           freq_server       ntpstats          Xorg.1.log
curator                 freq_server_dns   redis
analyst@SecOnion:/var/log$
```

D. Investigasi SQL *Injection Attack*

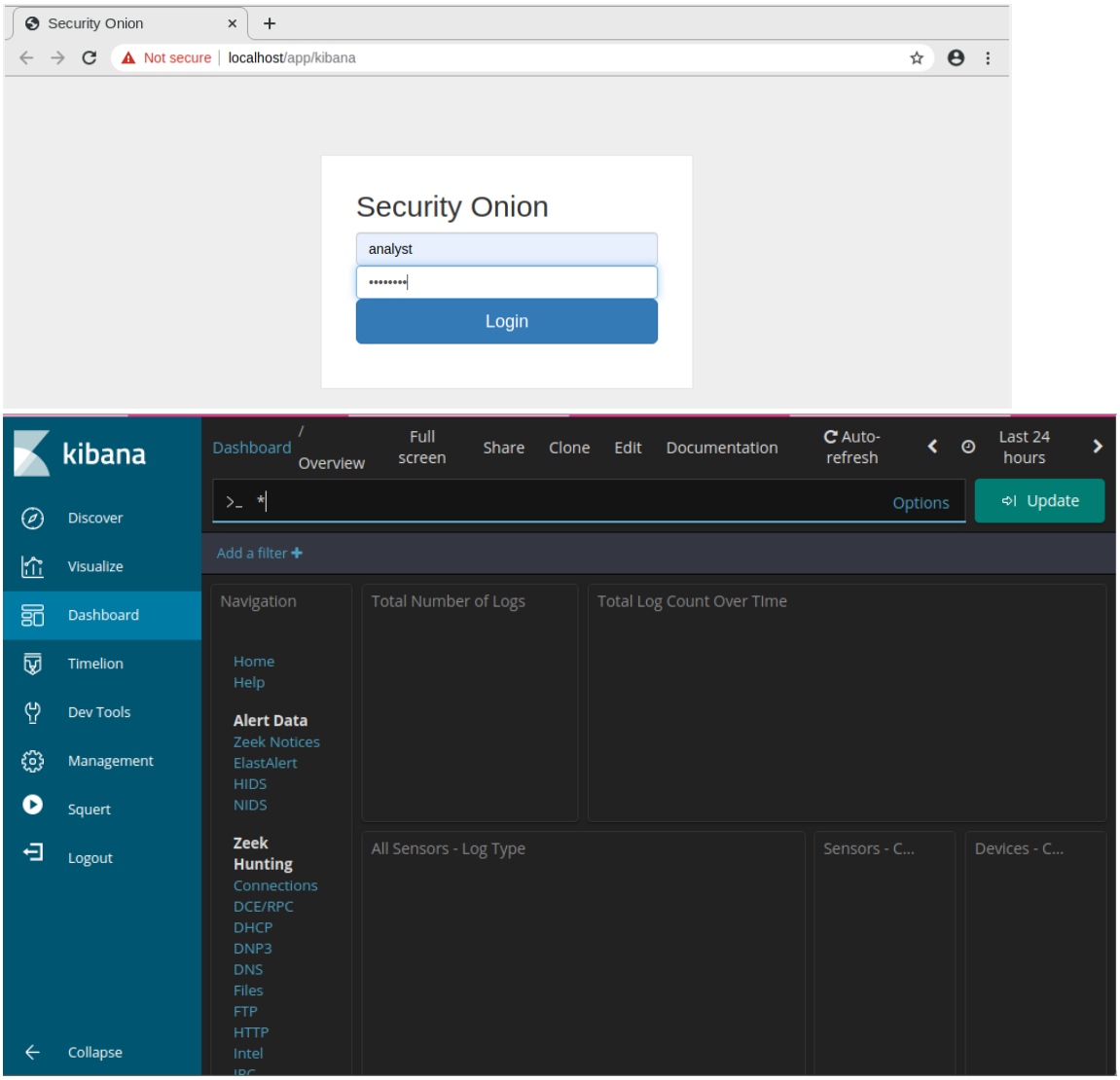
Langkah 1: Ubah Jangka Waktu/*Timeframe*

1. Masukkan perintah **sudo so-status** untuk memeriksa status layanan. Status untuk semua layanan harus OK sebelum memulai analisis. Ini bisa memakan waktu beberapa menit.

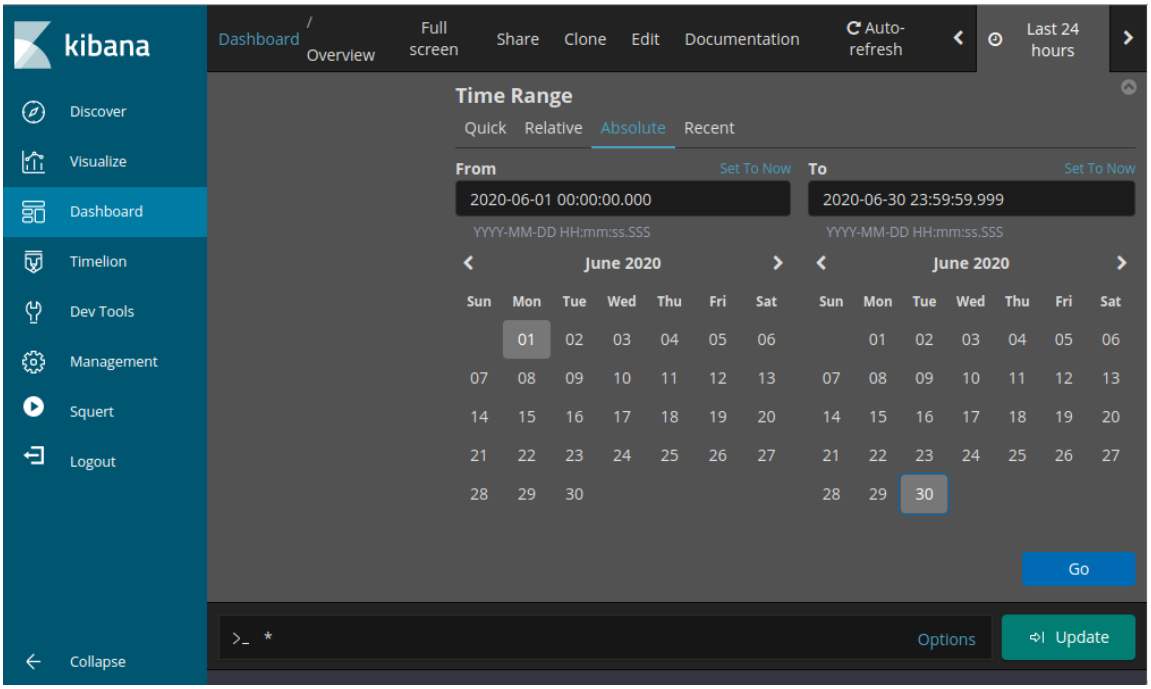
```
analyst@SecOnion:/var/log$ cd
analyst@SecOnion:~$ sudo so-status
[sudo] password for analyst:
Status: securityonion
  * sguil server                                                [  OK  ]
Status: seconion-import
  * pcap_agent (sguil)                                          [  OK  ]
  * snort_agent-1 (sguil)                                       [  OK  ]
  * barnyard2-1 (spooler, unified2 format)                      [  OK  ]
Status: Elastic stack
  * so-elasticsearch                                            [  OK  ]
  * so-logstash                                                 [  OK  ]
  * so-kibana                                                   [  OK  ]
  * so-freqserver                                               [  OK  ]

analyst@SecOnion:~$
```

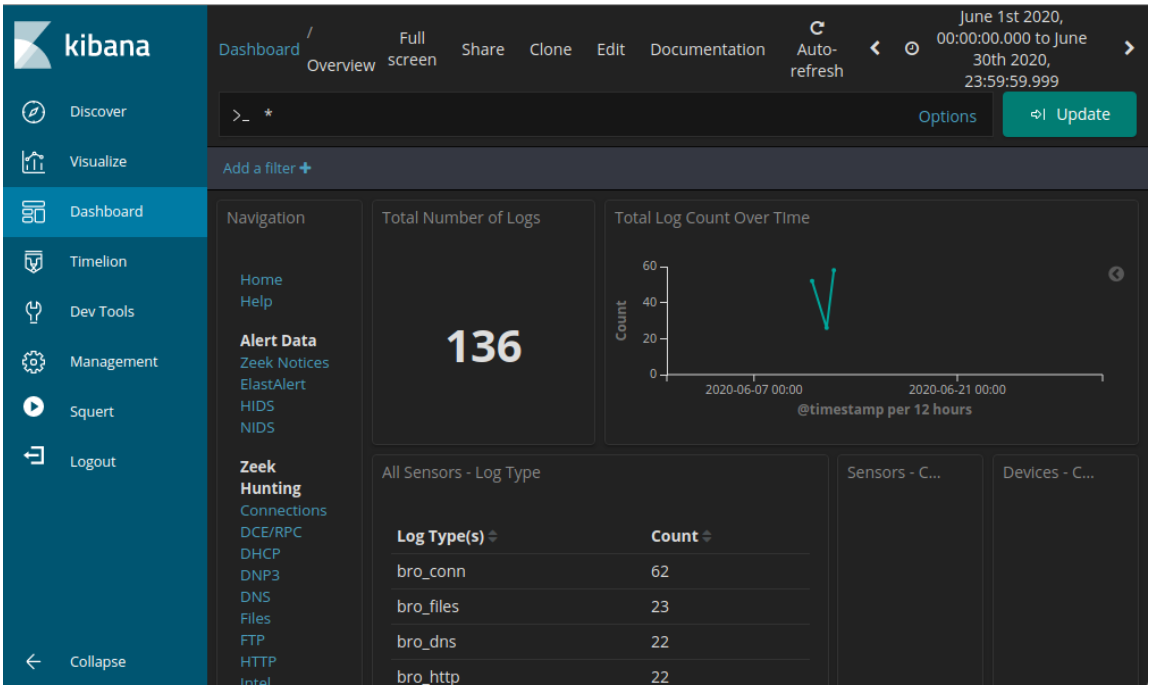
2. Setelah masuk, buka Kibana menggunakan pintasan di *Desktop*. Masuk dengan *username*: analyst dan *password*: cyberops.



3. Di sudut kanan atas jendela, klik 24 jam terakhir untuk mengubah ukuran *Time Range* sampel. Perluas *time range* untuk menyertakan peringatan yang menarik. Serangan injeksi SQL terjadi pada Juni 2020 jadi itulah yang perlu Anda targetkan. Pilih *Absolute* di bawah *Time Range* dan *edit* waktu *From* dan *To* untuk memasukkan seluruh bulan Juni di 2020. Klik *Go* untuk melanjutkan.

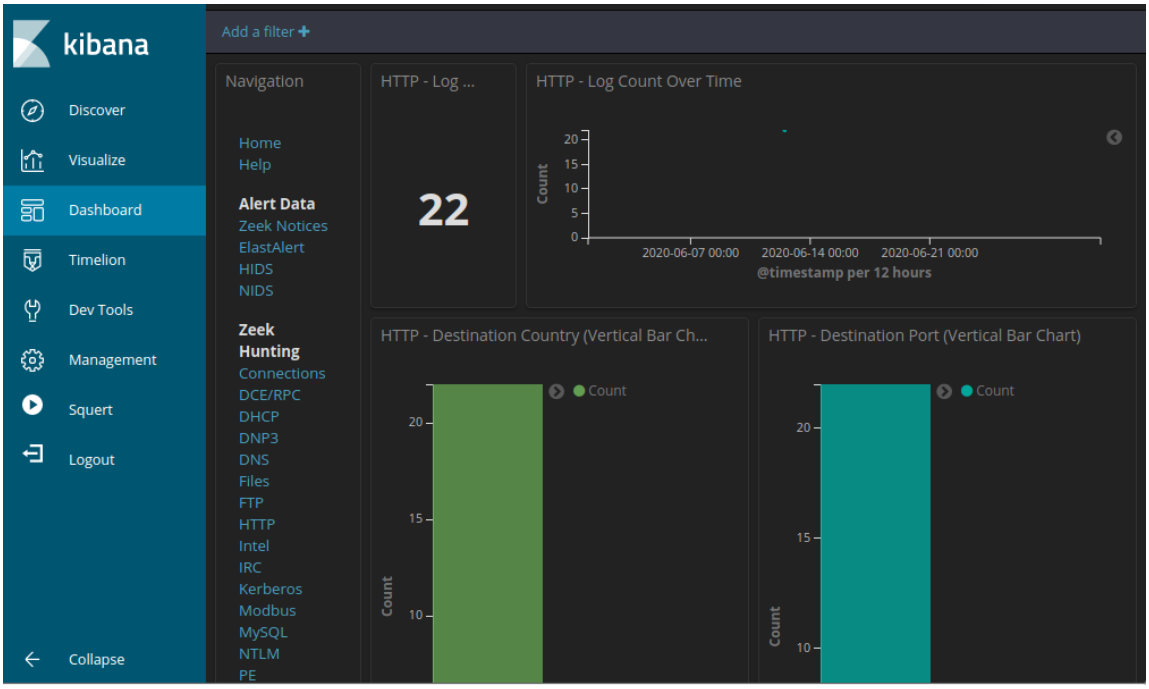


4. Perhatikan jumlah total *log* untuk seluruh bulan Juni 2020.

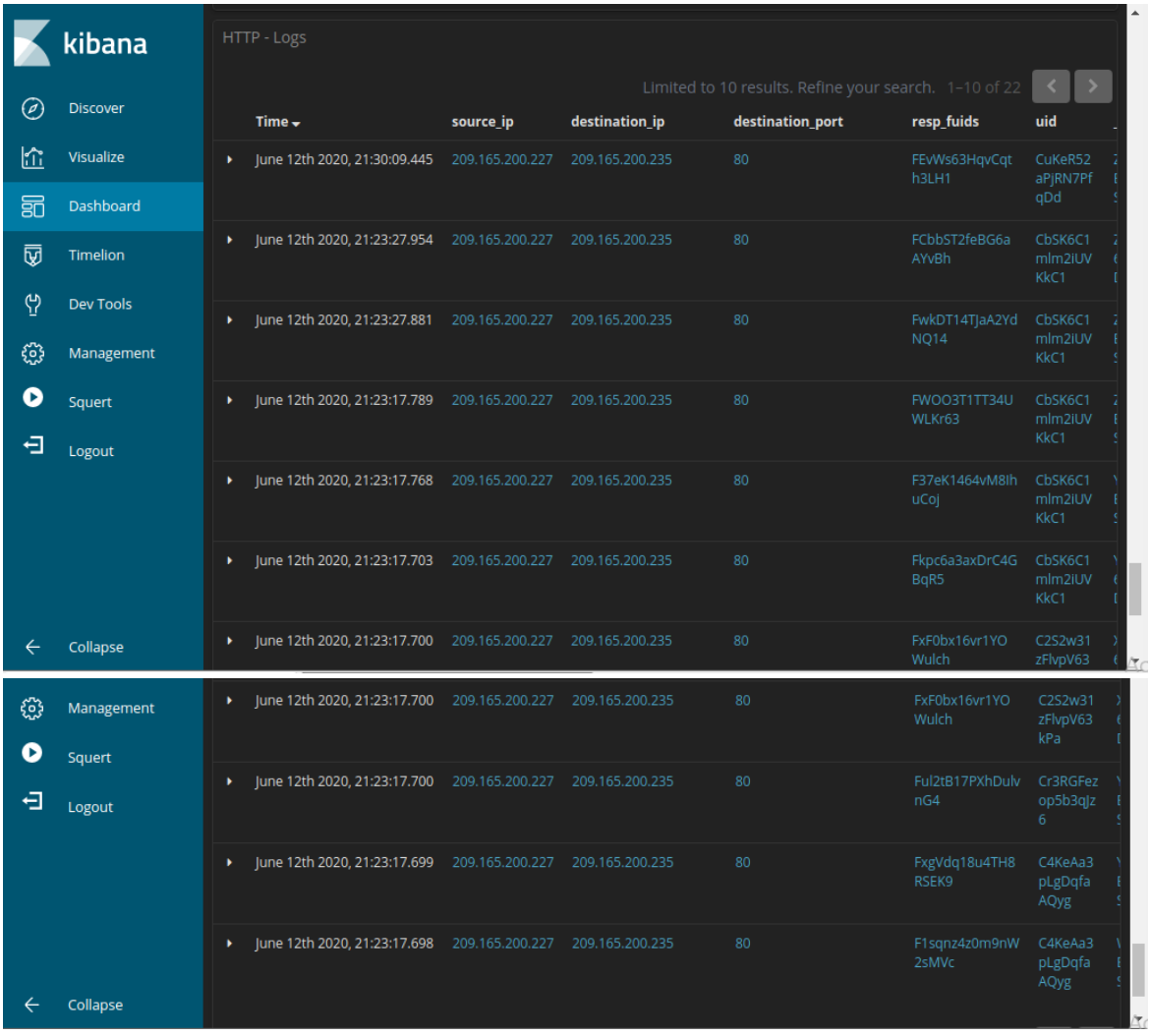


Langkah 2: Filter dari HTTP *Traffic*

5. Karena aktor ancaman menilai data yang disimpan di *server web*, *filter* HTTP digunakan untuk memilih *log* yang terkait dengan lalu lintas HTTP. Pilih HTTP di bawah judul Zeek *Hunting*, seperti yang ditunjukkan pada gambar.

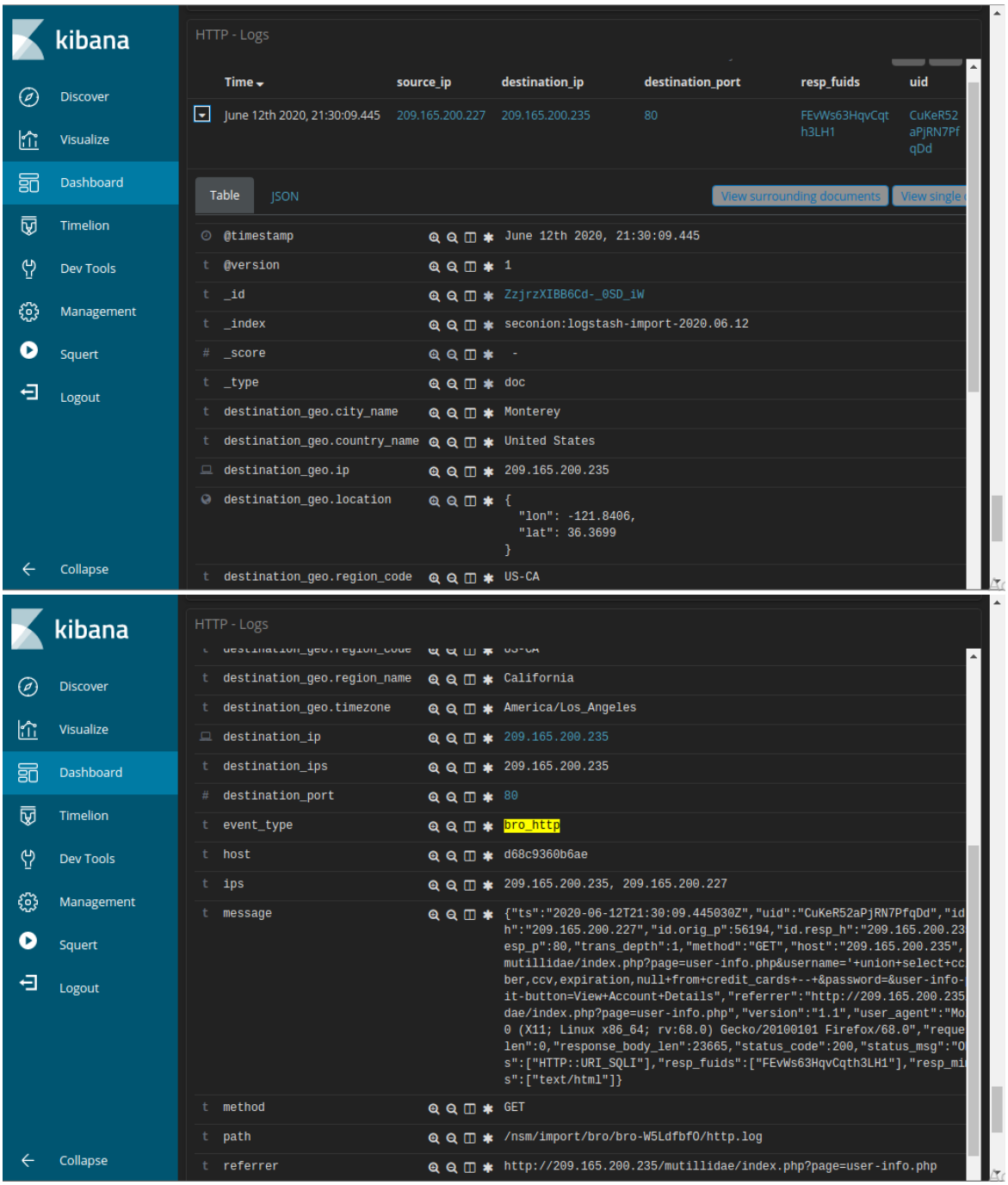


6. Gulir ke bawah ke *Log* HTTP. Lihat daftar 10 hasil pertama.



Dari hasil di atas, alamat IP sumber adalah 209.165.200.227 dan alamat IP tujuan adalah 209.165.200.235. Lalu nomor *port* tujuan adalah 80.

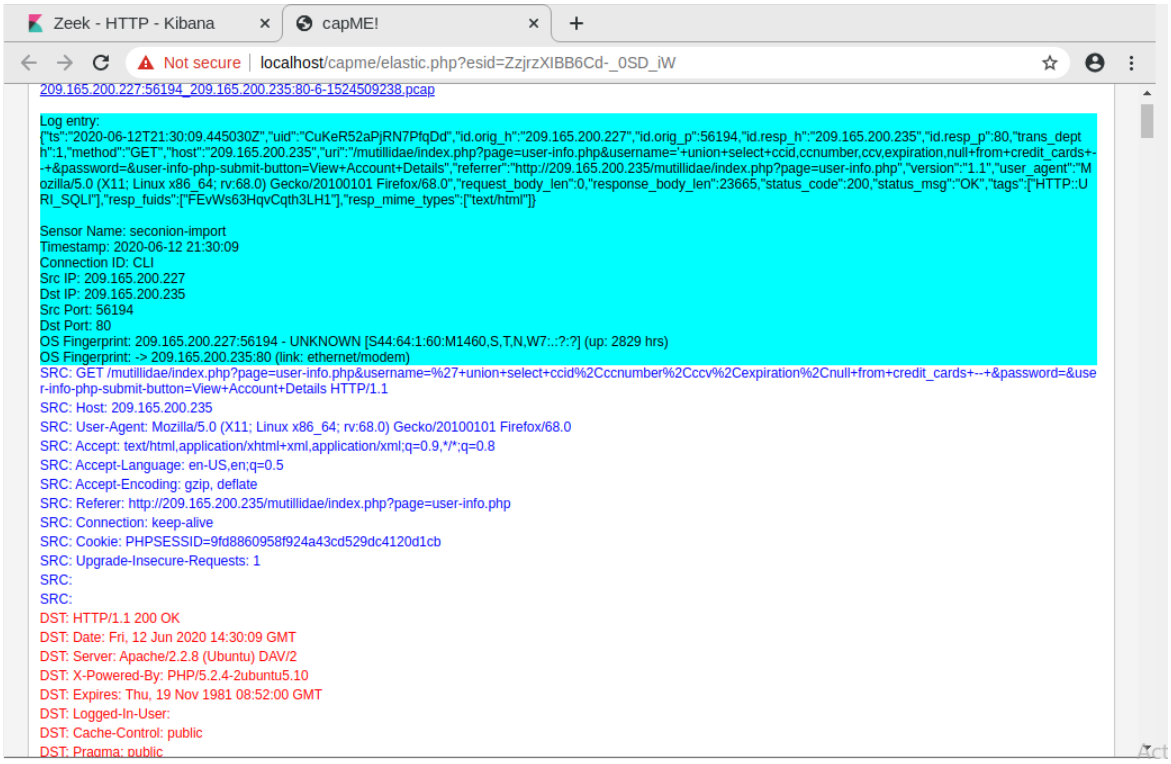
7. Klik detail hasil pertama dengan mengklik panah yang ada di sebelah *timestamp* entri *log*. Perhatikan informasi yang tersedia.



Langkah 3: *Review* Hasil

8. Beberapa informasi untuk entri *log* ditautkan ke alat lain. Klik nilai di bidang *alert* _id dari entri *log* untuk mendapatkan tampilan yang berbeda pada *event* tersebut.

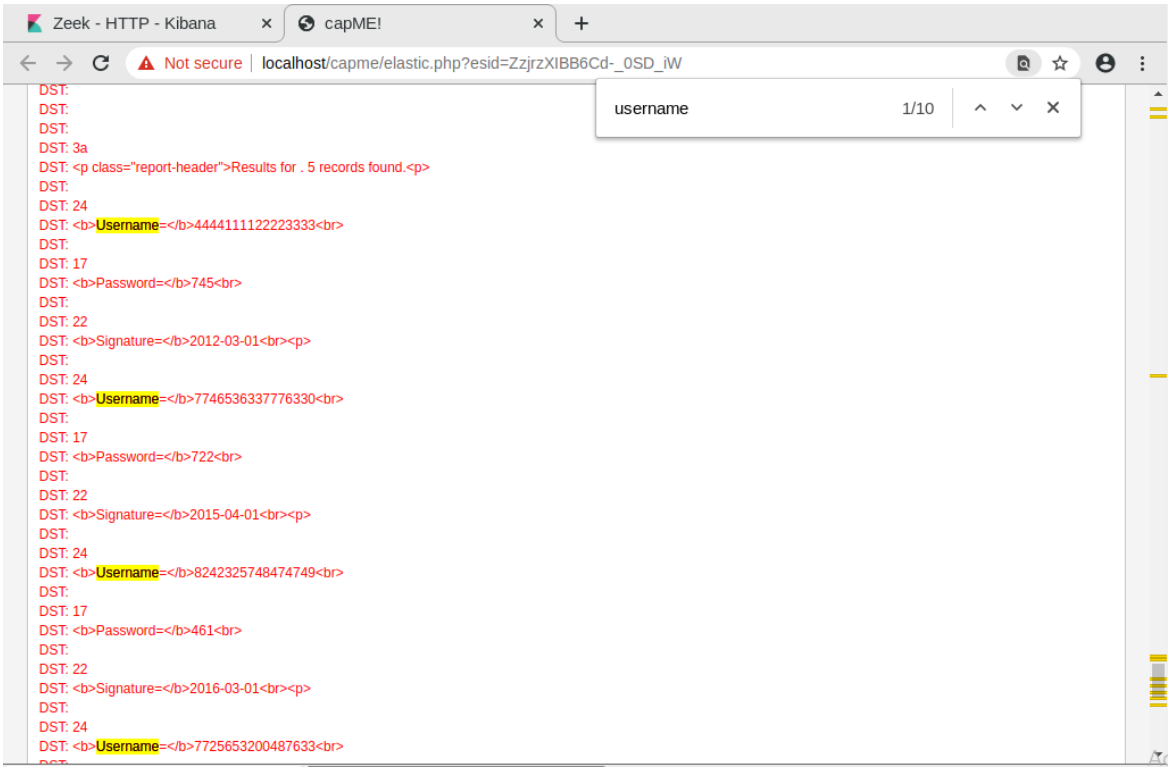
9. Hasilnya terbuka di *tab browser web* baru dengan informasi dari capME!.



10. Di bagian entri *Log*, yang ada di awal transkrip, perhatikan bagian **username='+union+select+ccid,ccnumber,ccv,expiration,null+from+credit_cards+--+&password=** menunjukkan bahwa seseorang mungkin telah mencoba untuk menyerang *browser web* menggunakan injeksi SQL untuk melewati otentikasi.

Log entry:
{"ts":"2020-06-12T21:30:09.445030Z","uid":"CuKeR52aPjRN7PfqDd","id.orig_h":"209.165.200.227","id.orig_p":56194,"id.resp_h":"209.165.200.235","id.resp_p":80,"trans_depth":1,"method":"GET","host":"209.165.200.235","uri":"/mutillidae/index.php?page=user-info.php&username='+union+select+ccid,ccnumber,ccv,expiration,null+from+credit_cards+--+&password=&user-info-php-submit-button=View+Account+Details","referrer":"http://209.165.200.235/mutillidae/index.php?page=user-info.php","version":"1.1","user_agent":"Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0","request_body_len":0,"response_body_len":23665,"status_code":200,"status_msg":"OK","tags":["HTTP::URI_SQLI"],"resp_fuids":["FEvWs63HqvCqth3LH1"],"resp_mime_types":["text/html"]}

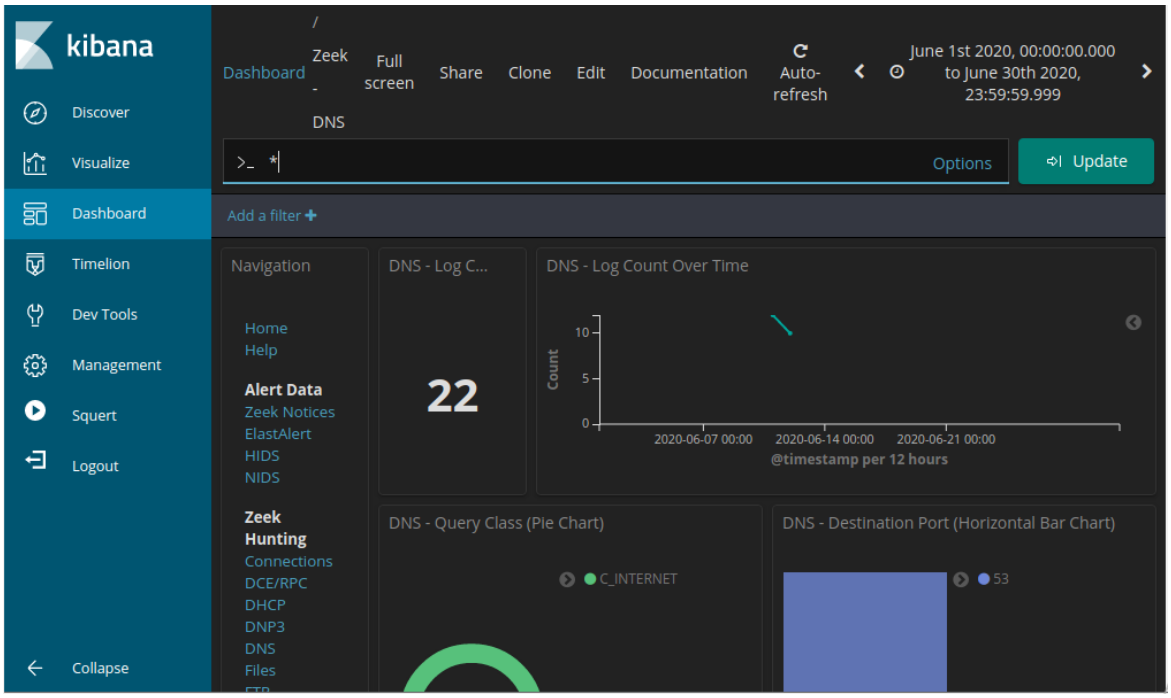
11. Temukan *keyword* nama pengguna dalam transkrip. Gunakan Ctrl-F untuk membuka kotak pencarian. Gunakan tombol panah bawah di kotak pencarian untuk menelusuri kejadian yang ditemukan.



Langkah 4: Analisis DNS *Exfiltration*

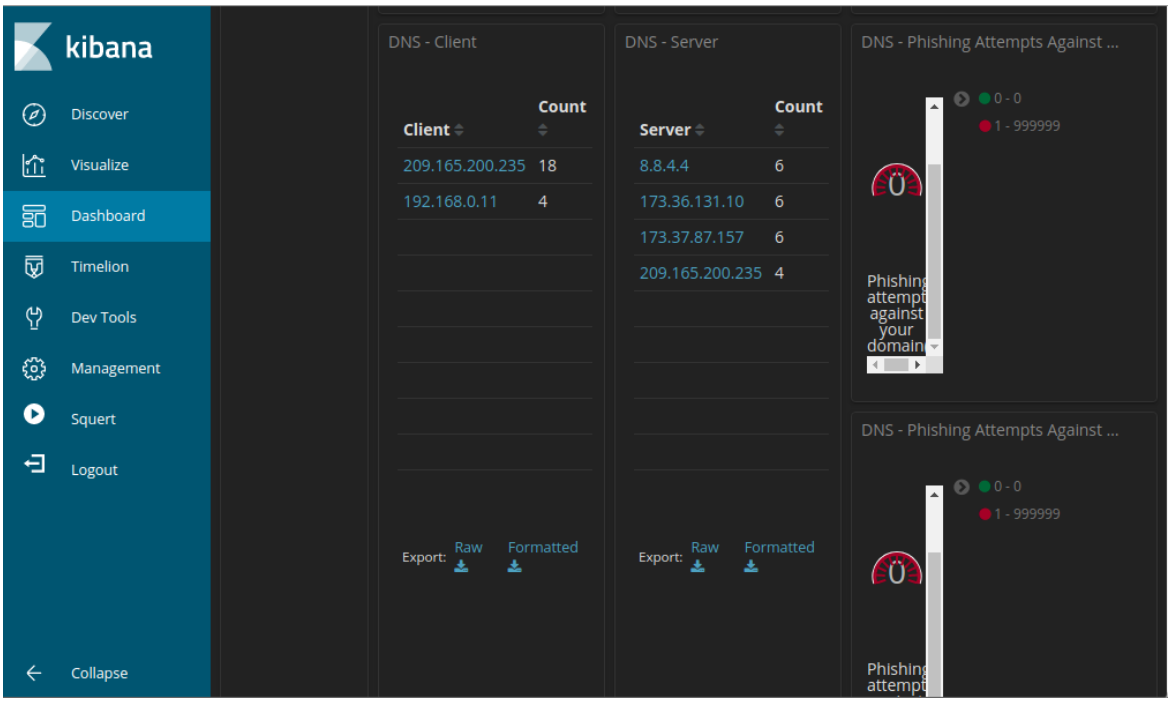
Filter DNS *Traffic*

12. Dari bagian atas Dasbor Kibana, hapus semua filter dan istilah pencarian dan klik Beranda di bawah bagian Navigasi Dasbor. Periode Waktu masih harus mencakup Juni 2020.
13. Di area *Dashboard* yang sama, klik DNS di bagian Zeek *Hunting*. Perhatikan metrik Jumlah *Log* DNS dan diagram batang horizontal *Port* Tujuan.

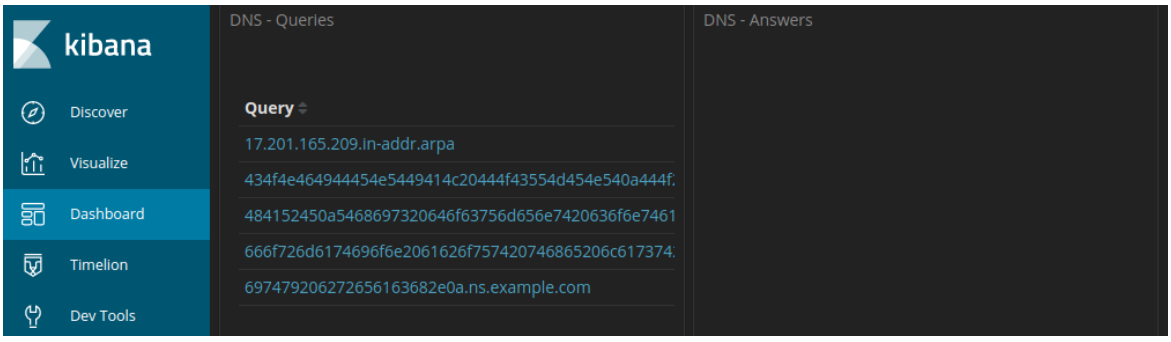


Tinjau Entri Terkait DNS

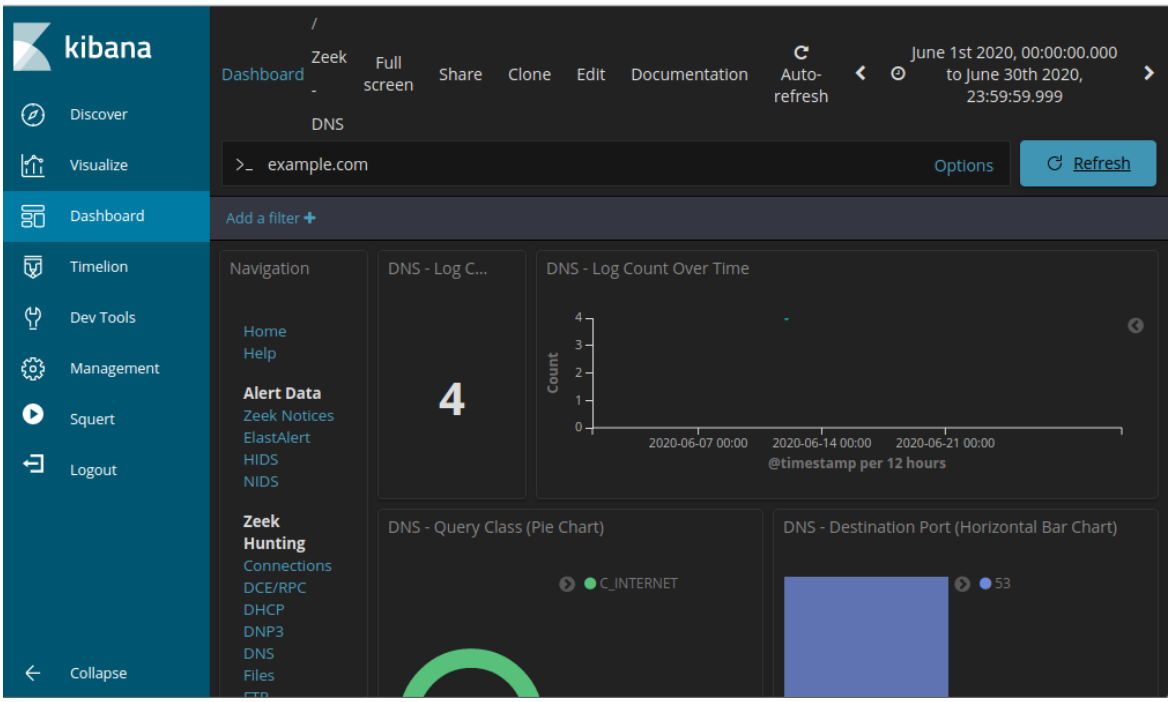
14. Gulir ke bawah jendela. Anda dapat melihat jenis kueri DNS teratas. Anda mungkin melihat catatan alamat (catatan A), alamat IPv6 catatan Quad A (AAAA), catatan NetBIOS (NB) dan catatan *pointer* untuk menyelesaikan nama *host* (PTR). Anda juga dapat melihat kode respons DNS.
15. Dengan Menggulir lebih jauh ke bawah, Anda dapat melihat daftar klien DNS dan *Server* DNS teratas berdasarkan jumlah permintaan dan respons mereka. Ada juga metrik untuk jumlah upaya DNS *Phishing*, yang juga dikenal sebagai *pharming* DNS, *spoofing*, atau *poisoning*.



16. Menggulir lebih jauh ke bawah jendela, Anda dapat melihat daftar kueri DNS teratas berdasarkan nama *domain*. Perhatikan bagaimana beberapa kueri memiliki *subdomain* yang sangat panjang yang dilampirkan ke ns.example.com. Domain example.com harus diselidiki lebih lanjut.

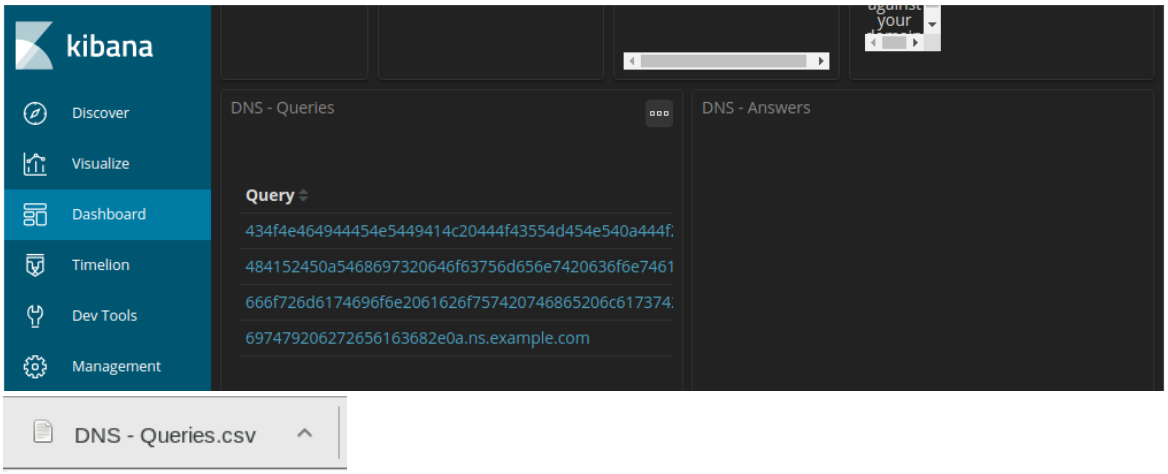


17. Gulir kembali ke bagian atas jendela dan masukkan example.com di bilah pencarian untuk memfilter example.com dan klik Perbarui. Perhatikan bahwa jumlah entri dalam Hitungan *Log* lebih kecil karena tampilan sekarang terbatas pada permintaan ke *server* example.com.



Tentukan Data yang Diekstraksi

18. Lanjutkan untuk menggulir lebih jauh ke bawah untuk melihat empat entri *log* unik untuk kueri DNS ke example.com. Klik tautan Ekspor: Unduh untuk mengunduh kueri ke file eksternal. *File* CSV diunduh ke *folder* /home/analyst/Downloads.



19. Arahkan ke *folder* /home/analyst/Downloads.

```
analyst@SecOnion:~$ cd /home/analyst/Downloads/
analyst@SecOnion:~/Downloads$ ls -l
total 4
-rw-rw-r-- 1 analyst analyst 304 Mar 14 03:23 DNS - Queries.csv
analyst@SecOnion:~/Downloads$
```

20. Di terminal, gunakan perintah xxd untuk memecahkan kode teks dalam *file* CSV dan menyimpannya ke *file* bernama secret.txt. Gunakan cat untuk menampilkan konten secret.txt ke konsol.

```
analyst@SecOnion:~/Downloads$ xxd -r -p "DNS - Queries.csv" > secret.txt
analyst@SecOnion:~/Downloads$ cat secret.txt
CONFIDENTIAL DOCUMENT
DO NOT SHARE
This document contains information about the last security breach.
analyst@SecOnion:~/Downloads$
```

V. Pembahasan

Bagian pertama adalah menganalisis *log* yang ditangkap dan pengambilan *traffic*. Pada kasus ini digunakan *file* 'nimda.download.pcap' yang berisi pengambilan paket yang terkait dengan unduhan *malware*. Kemudian sebagai contoh diambil tangkapan dengan HTTP. Karena HTTP berjalan di atas TCP, dimungkinkan untuk menggunakan fitur *Follow TCP Stream* Wireshark untuk membangun kembali transaksi TCP. Pada menu "*Follow TCP Stream*" di Wireshark, terdapat simbol dan informasi yang memberikan gambaran tentang data yang ditampilkan. Dari yang ditampilkan pada menu tersebut, dapat simpulkan bahwa klien mengirim permintaan HTTP GET untuk *file* "W32.Nimda.Amm.exe" ke *server* dengan alamat IP "209.165.202.133" pada *port* "6666". Permintaan ini juga mencakup informasi tambahan seperti *User-Agent, Accept-Encoding*, dan *Host*. *Server* merespons dengan kode status "200 OK", yang menunjukkan bahwa permintaan berhasil. *Server* mengirimkan *file* "W32.Nimda.Amm.exe" sebagai respons dengan menggunakan tipe konten "application/octet-stream". Informasi lainnya dalam respons termasuk *Server* yang digunakan (nginx/1.12.0), tanggal respons (*Date*), panjang konten (*Content-Length*), waktu modifikasi terakhir (*Last-Modified*), dan informasi lainnya seperti *Connection, ETag*, dan *Accept-Ranges*. Dalam tampilan "*Follow TCP Stream*" Wireshark, simbol ">" dan "<" digunakan untuk membedakan antara data yang dikirim dan diterima dalam komunikasi antara klien dan *server*. Simbol ">" menunjukkan data yang dikirim oleh *server*, sedangkan simbol "<" menunjukkan data yang diterima oleh klien.

Bagian kedua adalah *extract files* yang diunduh dari PCAP. Wireshark menampilkan semua objek HTTP yang ada dalam aliran TCP yang berisi permintaan GET, tetapi hanya objek-objek yang ditemukan dalam menu tersebut yang akan ditampilkan. Ketika melakukan ekspor objek HTTP melalui opsi "*Export Objects* > HTTP" di Wireshark, Wireshark akan mengekstrak semua objek yang sesuai dengan permintaan GET dari aliran TCP tersebut. Karena W32.Nimda.Amm.exe adalah satu-satunya *file* yang ditemukan, itulah mengapa *file* tersebut menjadi satu-satunya *file* yang ditampilkan dalam daftar objek HTTP yang dapat diekspor.

Bagian ketiga dilakukan pada *Security Onion* VM. Pada bagian ini hanya melihat *file log* yang dihasilkan oleh Zeek, *Snort*, dan *Various*.

Bagian keempat adalah Investigasi SQL *Injection Attack*, disini akan menyelidiki eksploitasi di mana akses tidak sah dibuat ke informasi sensitif yang disimpan di *server web*. Pada praktiknya digunakan Kibana untuk menentukan sumber serangan dan informasi yang diakses oleh penyerang. Bagian ini akan dibagi ke dalam 4 sub bagian yang berbeda.

Sub bagian pertama yaitu Ubah Jangka Waktu/*Timeframe.* Hal yang harus diperhatikan sebelum memulai analisis adalah status untuk semua layanan harus OK. Untuk mengubah *Time Range* pada kibana bisa dengan klik menu *'Last 24 Hours'* pada bagian *Dashboard*. Pada praktikum ini, *time range* diatur pada sepanjang bulan Juni 2020 yang memiliki *total log* sebanyak 136.

Sub bagian kedua adalah Filter dari HTTP *Traffic*. Karena aktor ancaman menilai data yang disimpan di *server web*, filter HTTP digunakan untuk memilih *log* yang terkait dengan lalu lintas HTTP. Dari hasil 10 *log* pertama HTTP dapat diketahui bahwa , alamat

IP sumber adalah 209.165.200.227 dan alamat IP tujuan adalah 209.165.200.235. Lalu nomor *port* tujuan adalah 80. Pada praktikum ini, sebagai contoh detailnya diambil dari hasil *log* pertama. *Log* pertama memiliki *timestamp* pada 12 Juni 2020 dengan waktu 21:30:09.445. *Event type* pada *log* ini adalah bro_http. Dari yang terlihat, pada kolom pesan berisi informasi seperti waktu, metode permintaan (dalam hal ini GET), URI, kode status respons, informasi tambahan seperti informasi pengguna, informasi peramban, dan lain sebagainya. Informasi yang terdapat pada *log* termasuk penting karena dapat digunakan untuk memantau aktivitas dan mendeteksi serangan keamanan pada aplikasi *web*, serta sebagai jejak aktivitas yang berguna untuk keperluan audit dan *compliance verification.* Melalui analisis *log* HTTP, organisasi dapat meningkatkan keamanan, kinerja, dan pengalaman pengguna aplikasi *web*.

Selain informasi yang dianalisis pada sub bagian kedua, pada sub bagian ketiga ini akan dilihat beberapa informasi tambahan seperti transkrip PCAP. Pada transkrip yang ditampilkan, terdapat bagian '**username='+union+select+ccid,ccnumber,ccv,expiration,null+from+credit_cards+--+&password=**' yang menandakan bahwa seseorang mungkin telah mencoba untuk menyerang *browser web* menggunakan injeksi SQL untuk melewati otentikasi. Jika kita mencari informasi terkait *username* pada transkrip, maka kita akan menemukan beberapa *username, password,* dan *signature* yang telah dicuri atau diekspos secara tidak sah dari suatu entitas atau sistem. Contoh dari *username, password*, dan *signature* tersebut adalah sebagai berikut.

No.	*Username*	*Password*	*Signature*
1.	4444111122223333	745	2012-03-01
2.	7746536337776330	722	2015-04-01
3.	82422325748474749	461	2016-03-01

Setelah tadi menganalisis *log* HTTP, pada sub bagian keempat ini akan dilakukan analisis DNS *exfiltration* pada *log* DNS. Informasi yang dapat dilihat pada *log* DNS yaitu seperti daftar *Query*, *Client*, dan *Server*. Alamat IP klien yang terdaftar pada *log* ini adalah 209.165.200.235 dan 192.168.0.11. Sedangkan untuk alamat IP *server* adalah 8.8.4.4, 173.36.131.10, 173.37.87.157, dan 209.165.200.235.

VI. Kesimpulan

1. *Log* memiliki informasi yang dapat digunakan untuk memantau aktivitas dan mendeteksi serangan keamanan aplikasi *web.*
2. Melalui analisis log HTTP, organisasi dapat meningkatkan keamanan, kinerja, dan pengalaman pengguna aplikasi *web*.
3. Menjaga kerahasiaan informasi sensitif seperti *username, password*, dan *signature* sangat penting.

VII. Daftar Pustaka

Dzikry. (2023). *Internet Protocol: Pengertian, Fungsi, Tugas dan Istilahnya.* Diakses pada 16 Maret 2023 dari https://masdzikry.com/internet-protocol/