

**ANALISIS KERENTANAN APACHE LOG4J PADA**

**CVE-2021-44228 TERHADAP ANCAMAN REMOTE ACCESS TROJAN DENGAN METODE PENETRATION TESTING EXECUTION STANDARD**

**SKRIPSI**

**MUHAMMAD NUR IRSYAD 1807422020**

**PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN**

**JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER**

**POLITEKNIK NEGERI JAKARTA**

**2022**



**ANALISIS KERENTANAN APACHE LOG4J PADA**

**CVE-2021-44228 TERHADAP ANCAMAN REMOTE ACCESS TROJAN DENGAN METODE PENETRATION TESTING EXECUTION STANDARD**

**SKRIPSI**

**Dibuat untuk Melengkapi Syarat-Syarat yang Diperlukan**

**untuk Memperoleh Diploma Empat Politeknik**

**MUHAMMAD NUR IRSYAD**

**1807422020**

**PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN**

**JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER**

**POLITEKNIK NEGERI JAKARTA**

**2022**

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Nur Irsyad

NIM : 1807422020

Jurusan : TIK – Teknik Informatika dan Komputer

Program Studi : TMJ – Teknik Multimedia dan Jaringan

Judul Skripsi : Analisis Kerentanan Apache Log4j Pada CVE-2021-44228

terhadap Ancaman Remote Access Trojan Dengan Metode

Penetration Testing Execution Standard

1. Menyatakan dengan sebenarnya bahwa skripsiini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya dari orang lain. Kutipan pendapat dan tulisan orang lain ditunjuk sesuai dengan cara-cara penulisan karya ilmiah yang berlaku.
2. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung cirri-ciri plagiat dan bentuk-bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanksi atas perbuatantersebut.
3. Depok, \_\_ \_\_\_\_ 2022
4. Yang membuat pernyataan,
5. Muhammad Nur Irsyad
6. NIM. 1807422020

LEMBAR PENGESAHAN

Skripsi diajukan oleh:

Nama : Muhammad Nur Irsyad

NIM : 1807422020

Jurusan : TIK – Teknik Informatika dan Komputer

Program Studi : TMJ – Teknik Multimedia dan Jaringan

Judul Skripsi : Analisis Kerentanan Apache Log4j Pada CVE-2021-44228

terhadap Ancaman Remote Access Trojan Dengan Metode

Penetration Testing Execution Standard

1. Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari \_\_, tanggal \_\_, bulam \_\_\_\_, tahun \_\_, dan dinyatakan **LULUS**.
2. Disahkan oleh:
3. Pembimbing I : Ariawan Andi Suhandana, S.Kom., M.T.I. ( . . . . . . . . . . )
4. Penguji I : Defiana Arnaldy, S.Tp., M.Si. ( . . . . . . . . . . )
5. Penguji II : Fachroni Arbi Murad, S.Kom., M.Kom. ( . . . . . . . . . . )
6. Penguji III : Asep Kurniawan, S.Pd., M.Kom. ( . . . . . . . . . . )
7. Mengetahui:
8. Jurusan Teknik Informatika dan Komputer
9. Ketua
10. Mauldy Laya , S.Kom., M.Kom.

NIP. 197802112009121003

KATA PENGANTAR

AA

1. Depok, \_\_ \_\_\_\_ 2022
2. Muhammad Nur Irsyad

**SURAT PERNYATAAN PERSETUJUAN PUBLIKASI**

SKRIPSI UNTUK KEPENTINGAN AKADEMIS

1. Sebagai sivitas akademik Politeknik Negeri Jakarta, Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Nur Irsyad

NIM : 1807422020

Jurusan : TIK – Teknik Informatika dan Komputer

Program Studi : TMJ – Teknik Multimedia dan Jaringan

1. Demi mengembangkan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul:
2. Analisis Kerentanan Log4Shell pada CVE-2021-44228 terhadap Ancaman Remote Access Trojan dengan Metode Penetration Testing Execution Standard
3. Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non Eksklusif ini Politeknik Negeri Jakarta Berhak menyimpan, mengalihmediakan / formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.. Demikian pernyatan ini saya buat dengan sebenarnya.
4. Depok, \_\_ \_\_\_\_ 2022
5. Yang membuat pernyataan,
6. Muhammad Nur Irsyad
7. NIM. 1807422020

ABSTRAK

AA

1. **Kata Kunci**: aaa

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

**BAB I**

PENDAHULUAN

* 1. Latar Belakang

Dalam dunia siber, potensi ancaman dapat muncul dikarenakan terdapatnya celah kerentanan pada suatu sistem maupun infrastruktur. Hal tersebut membuat sistem dapat diserang melalui berbagai perantara yang sesuai dengan bentuk celahnya. Masalah kerentanan ini yang lalu dieksploitasi oleh penyerang dengan landasan untuk manfaat pribadi dan berbagai macam faktor lainnya (Calín et al., 2020). Salah satu dampak ancaman siber, yaitu kebocoran data internal, disebabkan karena kerentanan sistem mnembuat suatu malware dapat tertanam di dalam sistem korban. Hal ini membuat penyerang dapat mengontrol sistem korban secara jarak jauh untuk mengambil aset serta informasi digital secara transparan terhadap supervisi pertahanan sistem korban (Yin & Khine, 2019).

Salah satu kasus ancaman siber yang muncul pada akhir November 2021 dengan penyebab yang serupa adalah kerentanan Log4Shell, yaitu istilah pada kerentanan library Apache Log4j terhadap serangan remote shell. Hal ini juga dikonfirmasi oleh Oracle pada 10 Desember 2021, yang menjelaskan bahwa kerentanan dengan referensi CVE-2021-44228 tersebut menyebabkan penyerang dapat mengontrol sistem korban melalui penyalahgunaan user input dalam fitur logging nya. Langkah awal ini digunakan untuk mengunduh dan menjalankan arbitary code dalam program Java. Adanya eksekusi payload tersebut nantinya dapat membangun koneksi remote secara penuh, baik dengan reverse shell maupun bind shell, tanpa ada autentikasi diantaranya (Khan and Neha, 2016; Apache, 2021; CVE, 2021; Oracle, 2021). Salah satu perusahaan global yang menggunakan library Apache Log4j, Cisco, memiliki lebih dari 60 produk serta fitur yang terpengaruh terhadap kerentanan tersebut. Hal ini didukung karena library Apache Log4j memiliki fleksibilitas dalam implementasinya di berbagai macam platform, seperti cloud service dan software development (Cisco, 2021).

Ancaman global tersebut terefleksikan pada status referensi CVE-2021-44228 yang merupakan satu-satunya kerentanan Apache Log4j dengan nilai Common Vulnerability Scoring System (CVSS) tertinggi, yaitu 10.0. Hal yang juga membuatnya berbeda dari kerentanan Apache Log4j lainnya adalah kerentanan tersebut menjadi pelopor untuk 3 kerentanan Apache Log4j yang baru dalam waktu kurang dari tiga minggu (26/11/2021 – 11/12/2021) (Apache, 2021). Walaupun kerentanan CVE-2021-44228 sudah diperbaiki pada versi Apache Log4j selanjutnya, efesiensi dan efektivitas eksploitasi kerentanan ini tetap dapat dimanfaatkan dari sisi penyerang sebagai media serangan yang kuat dan stabil.

Adapun berdasarkan uraian diatas, penelitian ini ditunjukkan untuk menganalisa ancaman kerentanan Apache Log4j pada referensi CVE-2021-44228 terhadap pengembangan eksploitasinya dengan pendekatan whitebox testing. Pengembangan dilakukan pada pengujian post exploitation menggunakan ancaman Remote Access Trojan secara persistence. Keseluruhan tahapan pengujian berbasiskan pada model Penetration Testing Execution Standard (PTES) sebagai lingkup panduan pengujian dan analisisnya (Dalalana and Zorzo, 2017). Tahap eksploitasi pengujian didasarkan pada serangan Remote Code Execution (RCE) dengan memanfaatkan JNDI Inection. Dua bentuk vektor serangan yang akan digunakan adalah Hands-on-Keyboard dan BadUSB, yang mana keduanya memanfaatkan miskonfigurasi aplikasi atau sistem, serta lemahnya validasi request input pengguna (Biswas *et al.*, 2018). Pengujian kemudian dikembangkan dengan menyisipkan backdoorke dalam sistem target untuk mempertahankan stabilitas akses, yang mana memanfaatkan kerentanan Apache Log4j sebagai komponen utamanya. Mitigasi yang diadaptasikan merujuk kepada pendekatan static analysis serta pemanfaatan program pemantuan dan konfigurasi internal sistem. Analisis keseluruhan pengujian dilakukan pada hasil eksploitasi dari pasca mitigasi, yang nantinya digunakan sebagai tolak ukur untuk mengetahui seberapa luas dan besarnya tingkat keberhasilan mitigasi terhadap ancaman tersebut(CEH, 2013; Muñoz and Mirosh, 2016; Kaushik *et al.*, 2021).

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan di atas, maka rumusan masalah dalam penelitian dapat dijabarkan sebagai berikut:

1. Bagaimana tahap rancang bangun instrumen pengujian dan integrasinya dengan Apache Log4j yang sesuai dengan referensi CVE-2021-44228?
2. Bagaimana analisa pengujian serta mitigasinya pada kerentanan Apache Log4j terkait ancaman Remote Access Trojan dalam lingkup white box testing dengan berbasiskan metode PTES?
3. Bagaiamana dampak kondisi sumber daya sistem pada pengujian terhadap ancaman Remote Access Trojan?

1.3 Batasan Masalah

Adanya pembatasan suatu masalah digunakan untuk menghindari potensi pelebaran pokok masalah dari lingkup yang seharusnya, sehingga dapat membuat penelitian lebih terarah untuk tercapainya tujuan dari penelitian ini. Beberapa batasan masalah dalam penelitian ini kemudian dijabarkan sebagai berikut:

1. Batasan dalam perancangan instrumen pengujian
2. Instrumen dirancang pada model arsitektur client-server secara lokal dengan memanfaatkan virtualisasi Docker container
3. Framework Java yang digunakan untuk membangun aplikasi utama pengguna dan penyerang adalah Maven, dengan *library* ApacheLog4j pada versi 2.14.1 dalam versi Java 8 yaitu 1.8.0\_181 dan 1.8.0\_321
4. Mesin komputer yang dipakai berbasiskan platform Linux, sehingga seluruh payload, program, serta skrip akan disesuaikan ke arah tersebut
5. Batasan dalam implementasi pengujian dan mitigasinya
6. Pengujian dilakukan dengan berbasiskan metode PTES dalam lingkup white box testing. Vektor serangan yang digunakan berlandaskan pada kerentanan Log4Shell, yaitu serangan Hands-on-Keyboard dan BadUSB. Hal yang membedakan diantaranya adalah pemanfaataan kerentanan tersebut dari perspektif penyerang serta target
7. Bentuk mitigasi mencangkup pendekatan deteksi ancaman, dengan implementasi *static analysis,* pemanfaatan program pemantauan serta konfigurasi internal sistem, serta analisis terhadap implementasi pembaharuan versi Apache Log4j pada 2.15.0, 2.16.0, dan 2.17.0
8. Proses pengujian dilakukan dalam 2 tahap, yaitu pra dan pasca adanya mitigasi, sehingga tergambarnya pencapaian yang dapat dianalisa besar tingkat keberhasilannya
9. Batasan dalam mengukur kondisi sumber daya sistem pada mesin target
10. Pemantauan sumber daya dilakukan pada 3 tahap pengujian. yaitu saat sistem dalam kondisi normal, pre mitigasi, dan pasca mitigasi
11. Parameter sumber daya yang diukur adalah CPU Utilization, CPU Time Consumption, Memory Occupation, Network Utilization, Disk Read & Write, dan User’s Activity

1.4 Tujuan dan Manfaat

Berdasarkan rumusan masalah, adapun tujuan serta manfaat yang ingin dicapaikan dalam pembentukan penelitian ini. Tujuan penelitian dijabarkan sebagai berikut:

1. Memberikan adanya suatu kontribusi pengembangan Proof-of-Concept (PoC) terhadap ancaman Apache Log4j pada CVE-2021-44228, terkhusus dalam pengembangan Remote Access Trojan
2. Menganalisis tingkat keberhasilan dari pengujian terhadap mitigasinya pada penggunaan attack vector Hands-on-Keyboard dan Bad USB dengan metode PTES secara dalam lingkup white box testing

Berdasarkan tujuan penelitian yang hendak dicapai, diharapkan pula adanya manfaat dari penelitian ini baik secara teoretis dan praktis, yaitu sebagai berikut:

1. Bagi masyarakat, penelitian ini diharapkan dapat memberikan wawasan terkait pentingnya kerentanan terhadap teknologi yang digunakan oleh pengguna, dan bagaimana dampak potensi dari ancaman serangannya
2. Bagi praktisi keamanan, penelitian ini diharapkan dapat memberikan sumbangan pemikiran pada analisis keamanan dalam dunia siber, serta sebagai dasar tambahan dalam mengkaji lebih lanjut terhadap kerentanan Apache Log4j pada referensi CVE-2021-44228 dan selanjutnya
3. Bagi penulis, penelitian ini digunakan sebagai bentuk implementasi dari pengembangan ilmu yang dipelajari selama masa kuliah di Politeknik Negeri Jakarta, serta diharapkan dapat memberikan kontribusi referensi kepustakaan keamanan siber pada lingkungan kampus hingga global

1.5 Sistematika Penulisan

1. **BAB I PENDAHULUAN**
2. Bab ini mendeskripsikan latar belakang serta urgensi masalah, perumusan masalah, batasan penelitian, tujuan & manfaat penelitian, serta struktur tulisan
3. **BAB II TINJAUAN PUSTAKA**
4. Bab ini membahas landasan teori yang digunakan dalam pembahasan penelitian dari sumber yang kredibel. Adapun penjabaran terkait penelitian sejenis sebagai penunjang dari penelitian sebelumnya dalam 10 tahun terakhir
5. **BAB III METODE PENELITIAN**
6. Bab ini memaparkan atribut inti dari penelitian, seperti metode yang digunakan dalam melakukan penelitian, tahapan dalam mendapatkan hasil pengujian dan analisanya, serta penjelasan singkat terhadap objek yang diteliti dalam laporan ini
7. **BAB IV HASIL DAN PEMBAHASAN**

Bab ini menjabarkan bagaimana tahapan dalam merancang, membangun dan mengimplementasikan instrumen pengujian, melakukan pengujian pada program dan kerentanan sistem, serta mengevaluasi dan menganalisa hasil pengujian

1. **BAB V PENUTUP**

Bab penutup menjelaskan mengenai pembuktian terhadap tujuan yang ingin dicapai dalam penelitian dan bagaimana hasil penelitiannya. Adapun saran yang diberikan terkait dengan hasil pengujian yang sifatnya konstruktif

**BAB II**

TINJAUAN PUSTAKA

2.1 Remote Access Trojan

1. Trojan dalam lingkup siber dapat diartikan sebagai medium untuk bagaimana serangan malware dikemas sedemikian rupa agar serangan tetap bersifat false negative terhadap sistem keamanan. Payload trojan dapat dikirim menggunakan berbagai macam pendekatan, seperti phishing, adware, ataupun dengan social engineering. Berdasarkan cara serangnya, tipe Remote Access Trojan (RAT) dispesifikasikan untuk mengontrol sistem korban sepenuhnya secara jarak jauh, atau remote, yang memanfaatkan koneksi berarsitektur client-server di antaranya. Pendekatan ini dimanfaatkan oleh penyerang untuk dapat mengontrol aset serta resource korban untuk dikelola sepenuhnya secara kontinuitas (CEH, 2013; Hama Saeed, 2020). Dalam membangun remote access, keberhasilan serta stabilitas koneksi bergantung kepada topologi infrastruktur jaringannya, terutama terhadap peranan firewall (Maraj, Rogova and Jakupi, 2020). Secara umum, terdapat 2 bentuk payload yang dapat digunakan untuk melakukan remote access, yaitu secara reverse dan bind, yang mana keduanya ditunjukkan untuk mengontrol sistem korban melalui akses shell yang didapatkannya.

2.1.1 Reverse & Bind Shell TCP

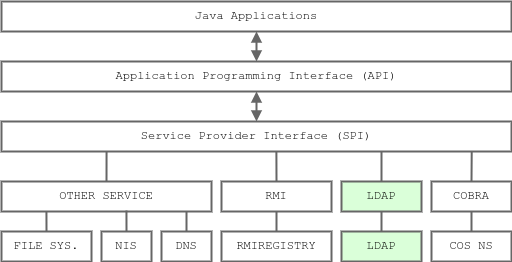
1. Bind shell bekerja dengan membuka layanan koneksi TCP di mesin korban pada port tertentu, yang juga disebut sebagai listener. Koneksi tersebut kemudian disambungkan oleh mesin penyerang untuk mendapatkan shell korban melalui remote access nya. Dikarenakan listener dilakukan dari mesin korban, hal ini harus disesuaikan dengan inbound rules yang mungkin terdapat dalam firewall, baik berupa eksternal maupun firewall sistem, sehingga koneksi listener dapat berfungsi sebagaimana harusnya (Saroeval and Bhadola, 2022).

Berbeda dengan payload bind shell, reverse shell bekerja dengan membuat listener dari mesin penyerang, lalu membutuhkan sistem korban untuk menyambungkan koneksi tersebut. Pendekatan ini akan merendahkan potensi isu terkait peranan firewall. Hal ini

disebabkan karena koneksi yang keluar dari mesin korban, atau outbound connection, memiliki kontrol yang lebih longgar daripada inbound connection pada firewall, sehingga sistem akan menanggap komunikasi tersebut sebagai koneksi yang valid dari sistem korban (Maraj, Rogova and Jakupi, 2020).

2.2 Apache Log4j

1. Apache Log4j merupakan framework Java yang umum digunakan untuk mengaudit berbagai macam pesan error hinnga info debug, baik pada perangkat lunak, jaringan, hingga layanan cloud computing (Rajasinghe, 2022). Dalam melakukan fungsinya, Apache Log4j juga dapat terintegrasi dengan berbagai macam layanan naming and directory untuk mencari dan mengambil objek data di dalamnya. Hal ini dilakukan melalui penggunaan Java Naming and Directory Interface (JNDI). Pencarian objek dalam suatu layanan, atau fungsi lookup, dapat JNDI lakukan baik dalam lingkup remote ataupun lokal (Apache, 2022).



1. **Gambar 2.1** Arsitektur JNDI

**Sumber**: Roy, 2015

1. Pada gambar 2.1 di atas merupakan arsitektur dari penggunaan JNDI dalam suatu aplikasi Java. JNDI terdiri dari dua komponen utama, yaitu JNDI Application Programming Interface (API), serta JNDI service Provider Interface (SPI). JNDI SPI merupakan suatu mekanisme agar konektivitas layanan naming and directory dapat tersedia pada aplikasi secara dinamis. Konektivitas tersebut yang kemudian digunakan oleh Apache Log4j untuk mengakses informasi serta objek di dalam layanan tersebut menggunakan modul dari JNDI API. Sala satu layanannya yaitu Lightweight Directory Access Protocol (LDAP) (Roy, 2015).
2. 2.2.1 Lightweight Directory Access Protocol
3. LDAP merupakan salah satu layanan dengan arsitektur client-server yang berbasiskan struktur direktori dalam melakukan penyimpanan informasi. Bentuk konfigurasinya menggunakan format file tersendiri, yaitu LDAP Data Interchange Format (LDIF), yang berisikan skema suatu direktori informasi. Penggunaan beberapa skema LDIF secara terpisah dapat membantu dalam mendesain dan mempopulasi data agar lebih tergorganisir (Helmke, Hudson and Hudson, 2019).
4. Dalam penyimpanan datanya, LDAP menggunakan suatu object yang berisikan koleksi atribut dalam mendefinisikan suatu entri pada skema, yang disebut sebagai object class. Object class pun dapat dilakukan pewarisan atau inheritence, baik bersifat abstrak ataupun struktural, sehingga penggunaan child objcet class dapat mereferensikan atribut parent object class nya (Oracle, 2010). Berikut pada tabel 2.1 merupakan contoh pewarsian pada atribut dalam object class inetOrgperson:
5. **Tabel 2.1** Atribut pewarisan object class inetOrgPerson

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Atribut** | **Deskripsi** | **Object Class Pewaris** |
| 1 | uid | ID unik pengguna | top (user) |
| 2 | description | informasi entri | person |
| 3 | inetUserStatus | status keaktifan akun | inetUser |
| 4 | ou | nama unit organisasi | organizationalPerson |
| 5 | mail | Alamat email pengguna | - |

**Sumber**: Oracle, 2010

1. 2.2.2 Kerentanan CVE-2021-44228
2. Kerentanan Log4Shell, referensi CVE-2021-44228, secara resmi dipublikasikan oleh Apache pada 10 Desember 2021 bahwa seleuruh versi Apache Log4j rentan terhadap serangan RCE, yaitu pada versi 2.0-beta9 hingga 2.14.1. Publikasi ini disertakan dengan bentuk mitigasi yang ditawarkan, yaitu perilisan versi 2.15. Kerentanan ini dapat dikatagorikan sebagai zero-day vulnerability, dikarenakan eksploitasi ditemukan terlebih dahulu oleh publik sebelum praktisi keamanan. Besarnya pengaruh kerentanan disebabkan karena tingginya estimasi paket dalam repositori Java yang ketergantungan dengan library Apache Log4j tersebut.
3. Secara garis besar, eksplotasi dilakukan dengan menginjeksi pesan khusus pada sistem logging Apache Log4j, yang kemudian pesan tersebut akan diinterpretasi dan mengeksekusi perintah apapun di dalamnya. Log4Shell sendiri berfokuskan pada pemanfaatan layanan LDAP serta RMI secara remote, yang mana dirancang khusus oleh penyerang. Kedua layanan tersebut cocok digunakan karena mampu untuk menyimpan referensi payload object Java, berupa file class yang siap dijalankan oleh fungsi lookup JNDI (Hiesgen *et al.*, 2022; Rajasinghe, 2022). Berikut merupakan contoh pesan yang dapat digunakan beserta integrasinya dengan JNDI dan layanan LDAP untuk eksploitasi:

${jndi:ldap://domain-penyerang.com/Payload.class}

2.3 White Box Testing

1. White box testing merupakan salah satu pendekatan dalam suatu pengujian yang pengujinya memiliki seluruh informasi, akses kontrol, ataupun kendali terhadap pengembangan lingkungan pengujian. Pendekatan ini disebut juga sebagai full-knowledge test. White box testing umum digunakan pada tiga tujuan utama, yaitu untuk kebutuhan introspeksi, stabilitas, serta ketelitian terhadap objek pengujian. Dalam pengujian kerentanan, diharapkan pendekatan ini dapat mengetahui dan mendeteksi potensi adanya kerusakan tambahan, atau collateral damage, dalam sistem terhadap suatu kerentanan (Midian, 2002; Madhavi, 2016).

2.4 Penetration Testing Execution Standard

PTES merupakan salah satu framework yang tersedia dalam menjalankan evaluasi keamanan dengan berstandar bisnis dan industri yang komprehensif. Salah satu keunggulan PTES yaitu menyediakan panduan perencanaan yang konkrit untuk mendefinisikan bagaimana keseluruhan tahapan dapat dijalankan dengan benar (Dalalana and Zorzo, 2017). PTES terdiri dari 7 tahapan utama yang mencangkup seluruh kebutuhan dasar dalam menjalankan pengujian keamanan, yaitu:

1. Pre-Engagement: mendefinisikan lingkup instrumen pengujian, yang juga mencangkup waktu estimasi pengerjaan, objek yang diteliti, serta tujuan utama dari pengujian
2. Intelligence Gathering: mengumpulkan kelengkapan informasi yang berkaitan dengan karakterisitik objek pengujian, baik dilakukan secara pasif maupun aktif
3. Threat Modelling: menggambarkan bagaimana ancaman dapat dilakukan berdasarkan dokumentasi kerentanan yang relevan, serta melakukan suatu pemetaan terhadap aset primer dan sekunder yang dapat ditargetkan
4. Vulnerability Analysis: menganalisis celah kerentanan untuk dapat mendefinisikan jalur serangan yang efektif serta lingkungan pengujiannya untuk dipersiapkan pada tahap eksploitasi
5. Exploitation: melakukan eksploitasi berdasarkan skema dan tujuan yang sudah dirancang sebelumnya, sehingga keakuratan informasi yang telah didapatkan akan mempengaruhi keberhasilan tahap eksploitasi
6. Post-Exploitation: mengembangkan hasil eksploitasi untuk menjadikan serangan yang lebih konsisten untuk tujuan kontinuitas, menunjukkan seberapa jauh kerentanan dapat dieksploitasi
7. Reporting: mendokumentasikan seluruh tahapan dan hasil kegiatan secara strukturan dan informatif, yang juga mencangkup kesimpulan dan saran serta pendekatan mitigasinya (Ningsih, 2021; PTES, 2021)
8. 2.4.1 Common Vulnerability Scoring System
9. CVSS merupakan framework untuk menentukan karakterisitik dan tingkatan suatu kerentanan terhadap teknologi. Penilaian CVSS terbagi menjadi 3 grup utama, yaitu Base, Temporal, dan Environmental, yang setiap katagori memiliki metrik penilaiannya sendiri. Dalam implementasinya, penggunaan seluruh metrik grup dapat menspesifikasikan tingkat kerentanan yang lebih sesuai dan akurat dengan lingkungan skenario pengujiannya. (FIRST, 2019). Pada tabel 2.2 berikut merupakan parameter dari metrik pada grup Base dalam CVSS versi 3.1, tabel 2.3 untuk metrik grup Temporal, serta 2.4 untuk metrik grup Environmental:
10. **Tabel 2.2** Keterangan metrik grup Base pada CVSS versi 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Deskripsi** | **Metrik** | |
| Attack  Vector | adanya konteks mengenai bagaimana jangkauan eksploitasi dapat dilakukan sejauh mungkin | Network | N |
| Adjacent | A |
| Local | L |
| Physical | P |
| Attack  Complexity | kondisi yang harus dipenuhi agar eksploitasi dapat dilakukan | Low | L |
| High | H |
| Privilege  Required | pengaruh terhadap butuhnya tingkatan hak tertentu dalam menjalankan eksploitasi | None | N |
| Low | L |
| High | H |
| User  Interaction | kondisi dimana jalannya eksploitasi membutuhkan interaksi pengguna | None | N |
| Required | R |
| Scope | gambaran luasnya dampak eksplotasi diluar cangkupan area kerentanan | Unchanged | U |
| Changed | C |
| Confidentiality | besarnya akses terhadap aset sistem yang dapat dikelola apabila terjadi eksploitasi | High | H |
| Low | L |
| None | N |
| Integrity | tingkat kerusakan integritas pada aset sistem yang dapat dilakukan terhadap dampak eksploitasi | High | H |
| Low | L |
| None | N |
| Availability | besarnya sumber daya sistem serta layanan yang dapat dikontrol oleh penyerang melalui eksploitasi | High | H |
| Low | L |
| None | N |

**Sumber**: FIRST, 2019

1. **Tabel 2.3** Keterangan metrik grup Temporal pada CVSS versi 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Deskripsi** | **Metrik** | |
| 1. Exploit Code Maturity | mengukur seberapa tingginya status ketersediaan eksploitasi, bermacamnya teknik eksploitasi, serta keaktifan eksploitasi dalam sisi industri | Unproven | U |
| PoC | P |
| Functional | F |
| Not Defined | X |
|  |  | High | H |
| 1. Remediation Level | tingkat remediasi yang tersedia untuk publik, baik itu dari vendor secara langsung ataupun tidak tersedia sama sekali | Official Fix | O |
| Temp. Fix | T |
| Workaround | W |
| Not Defined | X |
| Unavailable | U |
| 1. Report Confidence | tingginya validasi laporan ataupun isu eksploitasi terhadap kerentanan, baik dalam bentuk publikasi ataupun penelitian | Unknown | U |
| Reasonable | R |
| Not Defined | X |
| Confirmed | C |

**Sumber**: FIRST, 2019

1. **Tabel 2.4** Keterangan metrik grup Environmental pada CVSS versi 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Deskripsi** | **Metrik** | |
| Security Requirement | 1. pengaruh kerentanan terhadap prinsip dasar keamanan aset dan layanan sistem, yaitu Confidentiality, Integrity dan Availability (CIA Triad) | Low | L |
| Medium | M |
| Not Defined | X |
| High | H |
| Modified   1. Base | adanya adaptasi nilai metrik pada grup Base yang disesuaikan kembali dengan lingkungan pengujian | | |

**Sumber**: FIRST, 2019

1. Dalam mengimplementasikan perumusan seluruh nilai metriknya, FIRST menyediakan kalkukalor CVSS versi 3.1 yang dapat diakses secara online pada halaman webnya. Nilai akhir setiap metrik grup dikemas dalam skala numerik, mulai dari tidak berbahaya sama sekali hingga pada status kritikal (FIRST, 2019).
2. 2.4.2 Attack Trees

Attack trees merupakan framework untuk menggambarkan bagaimana rangkaian pengujian dapat dilakukan, berdasarkan tujuan utama pengujian yang disusun seperti struktur pohon. Attack trees didasarkan pada perspektif penyerang dalam melakukan eksploitasi, dan meraih tujuan utamanya. Attack trees dapat memiliki beberapa sub tujuan, atau intermediate node, di dalam tujuan utamanya, yang disebut juga sebagai overall node. Dalam tingkatannya, setiap intermediate node memiliki leaf node untuk menggambarkan serangan yang dibutuhkan dalam meraih tujuan node tersebut, untuk melanjutkan ke node diatasnya. Setiap overall dan intermediate node dapat bersifat AND atau OR, yang digunakan untuk mendeskripsikan syarat suksesi dari tujuan suatu node terhadap komponen dibawahnya (Shevchenko *et al.*, 2018; Ingoldsby, 2021). Pada tabel 2.3 berikut merupakan simbol dan deskripsi dari komponen attack tree:

1. **Tabel 2.5** Deskripsi simbol attack trees

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | OR  Node | dibutuhkan dua atau lebih node yang sukses untuk dapat mencapai atau melanjutkan node yang ada diatasnya |
|  | AND  Node | hanya membutuhkan salah satu node yang sukses untuk mencapai atau melanjutkan node yang ada diatasnya |
|  | Leaf  Node | menggambarkan vektor serangan yang bersifat independen dan tidak dapat dijadikan sebagai node ataupun memilikinya |
|  | Line | menggambarkan relasi setiap komponen yang tersambung diantaranya |

1. **Sumber**: Ingoldsby, 2021
2. 2.4.3 Serangan Hands-on-Keyboard
3. Serangan Hands-on-Keyboard merupakan pendekatan yang terjadi di mana penyerang sudah berada di dalam lingkungan sistem target, lalu menggunakan media keyboard target untuk melakukan eksploitasi secara langsung. Selain dilakukan oleh penyerang, hal ini juga dapat dilakukan oleh pengguna sistem dengan memanfaatkan social engineering. Maka dari itu, pengontrolan keystroke pada tingkatan sistem ataupun aplikasi merupakan salah satu langkah dalam menghadapi ancaman siber yang mana menggunakan aktivitas pengguna secara langsung di dalamnya (LiveAction, 2022).
4. 2.4.4 Serangan BadUSB
5. BadUSB merupakan perangkat keras microcontroller yang ditunjukkan untuk mengemulasi perangkat Human Interface Device (HID) dalam sistem target, seperti keyboard, mouse, hingga pemindai sidik jari. Dikarenakan penggunaan perangkat HID tidak dilakukan pemindaian oleh sistem, tidak seperti perangkat eksternal hard drive ataupun flash drive, BadUSB dapat langsung menginjeksi payload ke dalam mesin target tanpa terdeteksi antivirus. Dalam halnya mengemulasi keyboard, dikarenakan serangan BadUSB bekerja di depan layar monitor, atau foreground, keseluruhan rangkaian injeksi keystroke akan pula tertampil. Kelemahan ini diminimalisir dengan kecepatan keystroke per huruf hingga milidetik untuk menyelesaikan seluruh injeksinya, sehingga durasi serangan dapat berkurang secara signifikan (Bojović *et al.*, 2019).

2.5 Unified Modelling Language

Unified Modeling Language (UML) merupakan pendekatan terhadap standarisasi visual dari skema pada suatu sistem, sehingga seluruh komponen dapat dijabarkan secara dinamis. UML juga dapat digunakan sebagai alat untuk menganalisa berbagai macam tingkatan dalam sistem, baik itu struktur aplikasi ataupun aktivitas penggunaan aplikasi. Contoh dua bentuk penggunaan UML tersebut adalah class diagram dan activity diagram (Sukic and Saracevic, 2012).

Class diagram merupakan salah satu bagian dari diagram struktur UML yang menggambarkan tingkatan class dan interface pada suatu aplikasi atau sistem. Pendekatan ini juga umum digunakan pada perancangan sistem dalam bahasa pemrograman berkelas tinggi, seperti Java dan C#, yang sama-sama berprinsip object-oriented, sehinga class diagram dapat menunjukkan komponen-komponen class seperti variabel, fungsi, serta dependensinya. (OMG, 2011b; Sukic and Saracevic, 2012). Berikut pada tabel 2.4 merupakan simbol dan keterangan yang digunakan pada class diagram:

1. **Tabel 2.6** Deskripsi simbol class diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | Class | klasifikasi karakterisitik objek |
| Atribut | 1. properti variabel dalam class |
| Operasi | 1. fungsi metode dalam class |
|  | Asosiasi &  Kardinal | 1. adanya relasi statis terhadap besarnya implementasi objek atau atribut dalam class lain, yang juga dinotasikan dengan kardinalitas |
|  | 1. ukuran terhadap berapa elemen pada class lain yang terasosiasi |

1. **Sumber:** OMG, 2011; Ismail, 2020
2. Berbeda dengan class diagram, activity diagram merupakan bagian dari diagram kegiatan UML yang menunjukkan alur kontrol suatu objek pada rangkaian kondisi dari suatu aktivitas. Salah satu tujuan utama penggunaan activity diagram yaitu menggambarkan bagaimana aktivitas sistem dapat dijalankan menggunakan berbagai macam sudut pandang komponen di dalamnya (OMG, 2011a; Ismail, 2020). Berikut pada tabel 2.5 merupakan simbol dan keterangan yang digunakan pada activity diagram:
3. **Tabel 2.7** Deskripsi simbol activty diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | Inisiasi | 1. node untuk memulai alur aktivitas |
| Final | 1. node untuk menyelesaikan alur aktivitas |
|  | Aksi | 1. aksi kegiatan dengan kata kerja, yang juga bisa digunakan untuk memanggil suatu operasi |
|  | Keputusan | 1. node untuk mengontrol keputusan alur aktivitas dengan memberikan keluaran benar dan salah |
|  | Sinyal  Kirim | 1. node untuk memberikan input untuk diproses pada aksi atau node selanjutnya |
|  | Sinyal  Terima | 1. node untuk menerima input yang datang untuk dilanjutkan ke aksi atau node selanjutnya |
|  | Partisi | 1. pemberian notasi terhadap alur kegiatan dengan karakterisitik yang sama, baik secara vertikal ataupun horizontal |

1. **Sumber:** OMG, 2011; Ismail, 2020

2.6 Penelitian Sejenis

1. Penyusunan laporan ini menggunakan referensi dari penelitian sebelumnya yang sejenis dan relevan dengan topik serta pembahasan penelitian terhadap studi kasus, yang digunakan untuk mengembangkan aspek analisis penelitian ini.
2. Penelitian Rajasinghe, Ravindu (2022) yang berjudul ‘Remote Code Execution Security Flaw in Apache Log4j2’, menganalisis eksploitasi kerentanan Apache Log4j yang berfokus pada CVE-2021-44228 terhadap serangan Remote Code Execution (RCE) dalam lingkup white box testing. Vektor serangan yang peneliti gunakan berupa JNDI Injection melalui HTTP header X-Api-Version. Bentuk akhir eksploitasi adalah didapatkannya reverse shell sisitem korban menggunakan program netcat. Adapun bentuk deteksi dan mitigasi yang diimplementasikan yaitu berupa static analysis, dengan pemeriksaan berkas log dan mematikan opsi lookup dalam konfigurasi Log4j (Rajasinghe, 2022).
3. Penelitian oleh Shita Widya Ningsih (2021) yang berjudul ‘Analisis Pengujian Kerentanan Situs Pemerintahan XYZ dengan PTES’, menganalisis rangkaian pengujian kerentanan dengan metode PTES terhadap situs web suatu lembaga pemerintahan dalam lingkup black box testing. Dengan metodologi pengujian, arah serta informasi setiap tahapanya dipaparkan secara terstruktur. Dari berbagai macam kerentanan yang didapatkan, peneliti melakukan eksploitasi pada ancaman dengan celah kerentanan yang terbesar, yaitu Reflected Cross Site Scripting (XSS) dan Clickjacking. Walaupun penelitian mengandung keseluruhan tahap PTES, tahap eksploitasi tidak ditunjukkan untuk mendapatkan akses remote dari sistem, sehingga serangan tidak dapat dikembangkan ke tahap post-exploitation. Bentuk mitigasi yang disarankan adalah penggunaan Web Application Firewall (WAF) serta pendekatan static analysis dengan mengamankan konfigurasi opsi header aplikasi serta filterisasi input pengguna (Ningsih, 2021).
4. Penelitian yang dilakukan Nanny, Prayudi serta Riadi (2019) dengan judul ‘Peningkatan Keamanan Data Terhadap Serangan Remote Access Trojan (RAT) pada Cybercriminal Menggunakan Metode Dynamic Static’, ditunjukkan untuk mensimulasikan bagaimana cara kerja serangan RAT beserta dengan mitigasinya dalam lingkup white box testing. Infrastruktur jaringan LAN dibangun dengan menggunakan 2 buah laptop untuk pengujian serta 2 buah router Mikrotik. Vektor serangan yang digunakan untuk mendistribusikan payload RAT nya, yaitu njRAT, adalah dengan memanfaatkan file sharing. Selain untuk deteksi ancaman, router Mikrotik digunakan untuk mengendalikan koneksi dengan memasang fungsi firewall untuk memblokir koneksi reverse shell pada port yang ditemukan. Penelitian ini juga diunggulkan dengan adanya analisis forensik pada file trojan serta koneksi tersebut. Analisis akhir kemudian dikemas pada komparasi sumber daya pada sistem korban pada sebelum diserang, saat diserang, serta penyerangan pasca mitigasi (Nanny, Prayudi and Riadi, 2019).

**BAB III**

METODE PENELITIAN

**BAB IV**

HASIL DAN PEMBAHASAN

**BAB V**

PENUTUP

DAFTAR PUSTAKA

Calín, M., Anchez, S. ´, Carrillo De Gea, J. M., Jos´, J., Luis, J., Fern´fernández-Alemán, F., Alemán, A., Jes´, J., Garcerán, J., Garcerán, G., & Toval, A. (2020). *Software Vulnerabilities Overview: A Descriptive Study, Tsinghua Science and Technology*. https://doi.org/10.26599/TST.2019.9010003

Yin, K. S., & Khine, M. A. (2019). Optimal Remote Access Trojans Detection Based on Network Behavior. *International Journal of Electrical and Computer Engineering*, *9*(3), 2177–2184. https://doi.org/10.11591/ijece.v9i3.pp2177-2184