



**ANALISIS KERENTANAN LOG4SHELL 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 LOG4SHELL 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 Gelar Sarjana Terapan 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 Log4Shell pada CVE-2021-44228 terhadap Ancaman Remote Access Trojan dengan Metode Penetration Testing Execution Standard

Menyatakan dengan sebenarnya bahwa skripsi ini 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.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung ciri-ciri plagiat dan bentuk-bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Depok, __ __ 2022

Yang membuat pernyataan,

Muhammad Nur Irsyad
NIM. 1807422020

LEMBAR PENGESAHAN

Skripsi diajukan oleh:

Nama : Muhammad Nur Irsyad
NIM : 1807422020
Program Studi : TMJ - Teknik Multimedia dan Jaringan
Judul Skripsi : Analisis Kerentanan Log4Shell pada CVE-2021-44228 terhadap Ancaman Remote Access Trojan dengan Metode Penetration Testing Execution Standard

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari __, tanggal __, bulam ____, tahun __, dan dinyatakan **LULUS**.

Disahkan oleh:

Pembimbing I : Ariawan Andi Suhandana, S.Kom., M.T.I. (.)
Penguji I : Defiana Arnaldy, S.Tp., M.Si. (.)
Penguji II : Fachroni Arbi Murad, S.Kom., M.Kom. (.)
Penguji III : Asep Kurniawan, S.Pd., M.Kom. (.)

Mengetahui:

Jurusan Teknik Informatika dan Komputer
Ketua

Mauldy Laya , S.Kom., M.Kom.
NIP. 197802112009121003

KATA PENGANTAR

Aaa

Depok, __ ____ 2022

Muhammad Nur Irsyad

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

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

Demi mengembangkan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul:

Analisis Kerentanan Log4Shell pada CVE-2021-44228 terhadap Ancaman
Remote Access Trojan dengan Metode Penetration Testing Execution Standard

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 pernyataan ini saya buat dengan sebenarnya.

Depok, __ __ 2022

Yang membuat pernyataan,

Muhammad Nur Irsyad

NIM. 1807422020

ABSTRAK

Aaa

Kata kunci: aaa

DAFTAR ISI

SURAT PERNYATAAN BEBAS PLAGIARISME.....	iii
lembar pengesahan.....	iv
KATA PENGANTAR.....	v
SKRIPSI UNTUK KEPENTINGAN AKADEMIS.....	vi
ABSTRAK.....	vii
DAFTAR ISI.....	viii
DAFTAR gambar.....	x
DAFTAR TABEL.....	xi
DAFTAR istilah.....	xii
pendahuluan.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat.....	4
1.5 Sistematika Penulisan.....	5
TINJAUAN PUSTAKA.....	6
2.1 Remote Access Trojan.....	6
2.1.1 Reverse & Bind Shell TCP.....	6
2.2 Apache Log4j.....	7
2.2.1 Lightweight Directory Access Protocol.....	8
2.2.2 Kerentanan CVE-2021-44228.....	8
2.3 White Box Testing.....	9
2.4 Penetration Testing Execution Standard.....	9
2.4.1 Common Vulnerability Scoring System.....	10
2.4.2 Attack Trees.....	13
2.4.3 Serangan Hands-on-Keyboard.....	13
2.4.4 Serangan BadUSB.....	14
2.5 Unified Modelling Language.....	14
2.6 Alpha Testing.....	17
2.7 Penelitian Sejenis.....	17
METODE PENELITIAN.....	19
3.1 Rancangan Penelitian.....	19
3.2 Tahapan Penelitian.....	19
3.3 Objek Penelitian.....	20
HASIL DAN PEMBAHASAN.....	21
4.1 Perancangan Sistem.....	21

4.1.1 Desain Topologi Jaringan.....	21
4.1.2 Desain Skema LDAP.....	21
4.1.3 Desain Class Diagram Aplikasi.....	21
4.2 Implementasi Sistem.....	21
4.2.1 Implementasi Sistem Pengguna.....	21
4.2.1.1 Instalasi dan Konfigurasi OpenLDAP Server.....	21
4.2.1.2 Pengembangan Aplikasi GUI Desktop LDAP Client.....	21
4.2.2 Implementasi Sistem Penyerang.....	21
4.2.2.1 Instalasi dan Konfigurasi OpenLDAP Server.....	21
4.2.2.2 Instalasi dan Konfigurasi Apache HTTP Server.....	22
4.2.2.3 Pengembangan Aplikasi Java HTTP Server.....	22
4.2.2.4 Pengembangan Aplikasi Go JSON REST API.....	22
4.2.2.5 Pengembangan Payload Java.....	22
4.2.2.6 Pengembangan BadUSB.....	22
4.3 Pengujian Aplikasi dan Sistem.....	22
4.3.1 Prosedur Pengujian Aplikasi.....	22
4.3.1.1 integration Testing.....	23
4.3.1.2 Unit Testing.....	23
4.3.2 Prosedur Pengujian Kerentanan Sistem.....	23
4.3.2.1 Pre-Engagement.....	23
4.3.2.2 Intelligence Gathering.....	24
4.3.2.3 Threat Modelling.....	24
4.3.2.4 Vulnerability Analysis.....	24
4.3.2.5 Exploitation.....	24
4.3.2.6 Post-Exploitation.....	24
4.3.2.7 Reporting.....	24
4.3.2.8 Post-Mitigation Exploitation.....	24
4.4 Hasil Pengujian Aplikasi dan Sistem.....	24
4.4.1 Evaluasi Hasil Pengujian Aplikasi.....	24
4.4.2 Evaluasi Hasil Pengujian Kerentanan Sistem.....	25
PENUTUP.....	26
5.1 Kesimpulan.....	26
5.2 Saran.....	26
DAFTAR pustaka.....	27

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR ISTILAH

Malware

Payload

Attack Vector

Library

Arbitrary Code

Shell

Framework

Siber

Proof-of-Concept

BAB I

PENDAHULUAN

1.1 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, kebocoran data internal, disebabkan karena kerentanan sistem membuat malware boleh 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 and 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 arbitrary 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 backdoor ke 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 merancang 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. Bagaimana 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
 - a) Instrumen dirancang pada model arsitektur client-server secara lokal dengan memanfaatkan virtualisasi Docker container
 - b) Framework Java yang digunakan untuk membangun aplikasi utama pengguna dan penyerang adalah Maven, dengan *library* Apache Log4j pada versi 2.14.1 dalam versi Java 8 yaitu 1.8.0_181 dan 1.8.0_321
 - c) Mesin komputer yang dipakai berbasiskan platform Linux, sehingga seluruh payload, program, serta skrip akan disesuaikan ke arah tersebut
2. Batasan dalam implementasi pengujian dan mitigasinya
 - a) 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 pemanfaatan kerentanan tersebut dari perspektif penyerang serta target

- b) Bentuk mitigasi mencakup 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
 - c) Proses pengujian dilakukan dalam 2 tahap, yaitu pra dan pasca adanya mitigasi, sehingga tergambarinya pencapaian yang dapat dianalisa besar tingkat keberhasilannya
3. Batasan dalam mengukur kondisi sumber daya sistem pada mesin target
 - a) Pemantauan sumber daya dilakukan pada 3 tahap pengujian. yaitu saat sistem dalam kondisi normal, pre mitigasi, dan pasca mitigasi
 - b) 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 dicapai 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 kepastakaan keamanan siber pada lingkungan kampus hingga global

1.5 Sistematika Penulisan

Struktur sistematika penulisan dapat dijabarkan sebagai berikut:

BAB I PENDAHULUAN

Bab ini mendeskripsikan latar belakang serta urgensi masalah, perumusan masalah, batasan penelitian, tujuan & manfaat penelitian, serta struktur tulisan

BAB II TINJAUAN PUSTAKA

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

BAB III METODE PENELITIAN

Bab ini memaparkan atribut inti dari penelitian, seperti metode yang digunakan dalam melakukan penelitian, tahapan dalam mendapatkan hasil pengujian dan analisisnya, serta penjelasan singkat terhadap objek yang diteliti dalam laporan ini

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

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

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

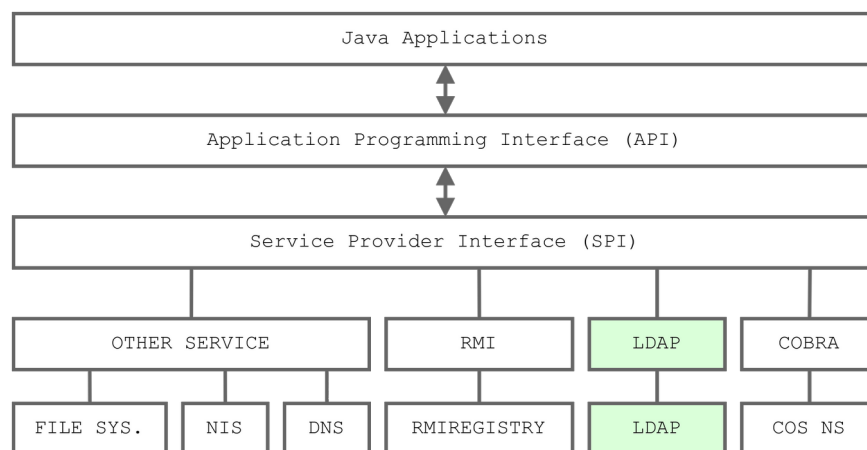
Bind shell bekerja dengan membuka layanan koneksi TCP di mesin korban pada port tertentu, yang juga disebut sebagai listener. Koneksi tersebut kemudian disambungkan oleh mesin penyerang untuk mendapatkan shell korban melalui remote access nya. Dikarenakan listener dilakukan dari mesin korban, hal ini 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 menanggapi komunikasi tersebut sebagai koneksi yang valid dari sistem korban (Maraj, Rogova and Jakupi, 2020).

2.2 Apache Log4j

Apache Log4j merupakan framework Java yang umum digunakan untuk mengaudit berbagai macam pesan error hingga 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).



Gambar 2.1 Arsitektur JNDI

Sumber: Roy, 2015

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. Salah satu layanannya yaitu Lightweight Directory Access Protocol (LDAP) (Roy, 2015).

2.2.1 Lightweight Directory Access Protocol

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 terorganisir (Helmke, Hudson and Hudson, 2019).

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 inheritance, baik bersifat abstrak ataupun struktural, sehingga penggunaan child object class dapat mereferensikan atribut parent object class nya (Oracle, 2010). Berikut pada tabel 2.1 merupakan contoh pewarisan pada atribut dalam object class inetOrgperson:

Tabel 2.1 Atribut pewarisan object class inetOrgperson

No.	Atribut	Deskripsi	Parent Object Class
1	uid	ID unik pengguna	top (user)
2	description	informasi entri	Person
3	inetUser	status keaktifan akun	InetUser
4	ou	nama unit organisasi	OrganizationalPerson
5	mail	alamat email pengguna	-

Sumber: Oracle, 2010

2.2.2 Kerentanan CVE-2021-44228

Kerentanan Log4Shell, referensi CVE-2021-44228, secara resmi dipublikasikan oleh Apache pada 10 Desember 2021 bahwa seluruh 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 dikategorikan 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.

Secara garis besar, eksploitasi dilakukan dengan menginjeksi pesan khusus pada sistem logging Apache Log4j, yang kemudian pesan tersebut akan diinterpretasi dan mengeksekusi perintah apapun di dalamnya. Log4Shell sendiri berfokus 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

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 mencakup seluruh kebutuhan dasar dalam menjalankan pengujian keamanan, yaitu:

1. Pre-Engagement: mendefinisikan lingkup instrumen pengujian, yang juga mencakup waktu estimasi pengerjaan, objek yang diteliti, serta tujuan utama dari pengujian
2. Intelligence Gathering: mengumpulkan kelengkapan informasi yang berkaitan dengan karakteristik 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 mencakup kesimpulan dan saran serta pendekatan mitigasinya (Ningsih, 2021; PTES, 2021)

2.4.1 Common Vulnerability Scoring System

CVSS merupakan framework untuk menentukan karakteristik 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:

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

Tabel 2.3 Keterangan metrik grup Temporal pada CVSS versi 3.1

Parameter	Deskripsi	Metrik	
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
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
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

Tabel 2.4 Keterangan metrik grup Environmental pada CVSS versi 3.1

Parameter	Deskripsi	Metrik	
Security Requirement	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 Base	adanya adaptasi nilai metrik pada grup Base yang disesuaikan kembali dengan lingkungan pengujian		



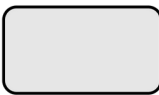

Sumber: FIRST, 2019

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).

2.4.2 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 di atasnya. 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:

Tabel 2.3 Deskripsi simbol attack trees

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

Sumber: Ingoldsby, 2021

2.4.3 Serangan Hands-on-Keyboard

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).

2.4.4 Serangan BadUSB

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).

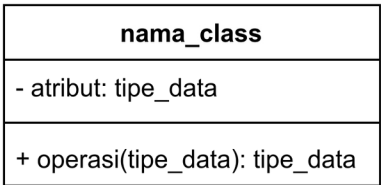
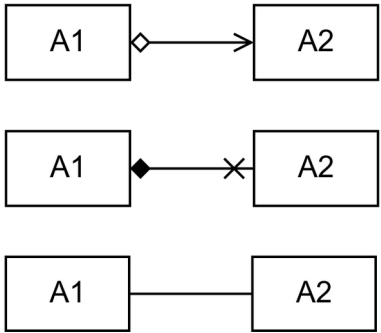
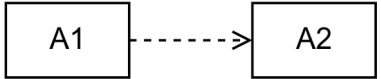
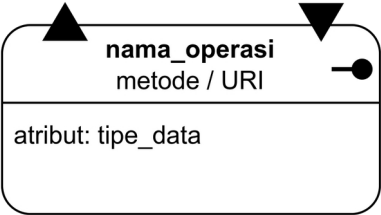
2.5 Unified Modelling Language

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, sehingga 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:


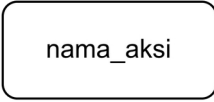

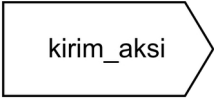
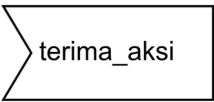
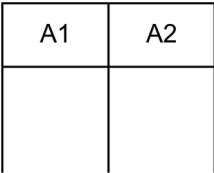
Tabel 2.4 Deskripsi simbol class diagram

Simbol	Nama	Deskripsi
	Class	klasifikasi karakteristik objek
	Atribut	properti variabel dalam class
	Operasi	fungsi metode dalam class
	Asosiasi & Kardinal	adanya relasi statis terhadap besarnya implementasi objek atau atribut dalam class lain, yang juga dinotasikan dengan kardinalitas
0..0 // 0..1 // 1..1 // 0..* // m..n		ukuran terhadap berapa elemen pada class lain yang terasosiasi
	Dependensi	adanya relasi abstrak terhadap referensi suatu elemen dalam class lain pada lingkup fungsi
	REST	proses pemanggilan fungsi dari layanan Representational State Transfer (REST) terhadap aplikasi

Sumber: OMG, 2011; Ismail, 2020

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:

Tabel 2.5 Deskripsi simbol activity diagram

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

Sumber: OMG, 2011a; Ismail, 2020

2.6 Alpha Testing

Pengujian aplikasi, atau software testing, ditunjukkan untuk mengevaluasi apakah kinerja aplikasi memenuhi spesifikasi tertentu. Pengujian aplikasi dikategorikan menjadi beberapa jenis, salah satunya adalah alpha testing. Pengujian ini termasuk sebagai bentuk acceptance testing, bersama dengan beta testing, yang merupakan pengujian sebelum aplikasi masuk ke dalam tahap produksi. Berbeda dengan beta testing yang pengujiannya berfokuskan kepada respon dari sampel pengguna secara langsung, alpha testing diuji oleh pengembang aplikasi tersebut untuk melakukan validasi dan verifikasi dari pemenuhan komponen atau modul. Hal ini ditunjukkan sebagai bentuk pemenuhan fungsional pada aplikasi secara utuh. Dikarenakan alpha testing dilakukan oleh pihak internal, maka pengujian dapat dikategorikan sebagai salah satu bentuk model white box testing serta grey box testing (Jamil *et al.*, 2017; Achmad and Yulfitri, 2020).

2.7 Penelitian Sejenis

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.

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 sistem 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).

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)

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

3.1 Rancangan Penelitian

Pendekatan yang digunakan pada penelitian ini adalah kuantitatif dengan jenis eksperimental. Dikarenakan pengujian serta perancangan instrumen pengujian dilakukan dalam lingkup white box, maka penggunaan batasan masalah yang diajukan digunakan sebagai variabel terkontrol. Hal ini dijaga agar hasil penelitian tidak terpengaruh dari faktor diluar aspek pengujian. Adapun teknik pengumpulan data penelitian yang difokuskan pada tipe sekunder, yaitu mencangkup referensi dari penelitian kepustakaan terdahulu serta studi dokumentasi baik dari sumber primer, seperti halaman web resmi vendor, maupun sekunder, seperti contoh PoC pengujian dari sumber terbuka. Dengan adanya data tersebut, peneliti kemudian dapat menguji serta menganalisis pengembangan permasalahan pada studi kasus ataupun penelitian terdahulu. Penelitian ini mencangkup 3 bentuk analisis utama, yaitu dalam pengembangan instrumen pengujian, pengerjaan prosedur pengujian dengan metode PTES beserta dengan mitigasinya, dan perbandingan signifikansi kondisi sumber daya sistem target terhadap hasil pengujian.

3.2 Tahapan Penelitian

Adapun tahapan-tahapan yang sifatnya prosedural dalam melakukan penelitian ini, yang dapat dijabarkan ke dalam beberapa poin utama sebagai berikut:

1. Perumusan Masalah

Peneliti mengumpulkan bahan literatur serta informasi terkait untuk mengidentifikasi masalah yang akan diangkat atau dikembangkan terhadap objek penelitian. Tahap ini juga digunakan untuk mendapatkan gambaran mengenai bagaimana bentuk pengujian serta analisisnya

2. Pengumpulan Data & Teori

Peneliti mengumpulkan informasi terkait dengan objek penelitian dari sumber yang kredibel, seperti bagaimana perancangan dan implementasi

lingkungan pengujiannya, teknologi apa yang digunakan dalam target aplikasi, dasar metode pengujian apa yang akan digunakan, serta seperti apa bentuk-bentuk mitigasi yang direkomendasikan. Informasi yang didapatkan tersebut lalu dirumuskan sebagai bentuk batasan masalah

3. Perancangan dan Pembangunan Instrumen Pengujian

Pada tahap ini peneliti mulai merancang serta membangun instrumen pengujian yang juga telah didasarkan pada rumusan batasan masalah. Instrumen penelitian mencakup lingkungan pengujian, sistem serta layanan yang akan digunakan, target aplikasi, serta program pengujiannya, seperti skrip payload dan program pendukung lainnya

4. Pengujian

Peneliti melakukan pengujian dalam dua tahap, yaitu menguji fungsional akhir dari instrumen pengujian, serta menguji kerentanan pada objek penelitian yang didasarkan pada metode PTES, menggunakan instrumen yang telah dibangun pada tahap sebelumnya

5. Analisis Hasil Pengujian

Selain menganalisis proses pada tahap sebelumnya, adapun dokumentasi pada hasil data pengujian akhir yang digunakan untuk mengukur besar dampak pengujian terhadap sistem target dari ancaman serangan melalui beberapa pengukuran sumber daya yang berbeda

3.3 Objek Penelitian

Objek yang diteliti dalam penelitian ini adalah kerentanan dari produk Apache Log4j terhadap ancaman serangannya pada referensi CVE-2021-44228. Dengan begitu, seluruh instrumen pengujian beserta dengan pengujiannya dibangun untuk dapat terintegrasi objek penelitian tersebut. Pada implementasi sesungguhnya, selain mengandalkan sistem target untuk memiliki kerentanan ini, objek penelitian kemudian dikembangkan untuk menjadi vektor serangan yang independen agar dapat mencapai tujuan yang sama, yaitu meraih tahap eksploitasi akhir melalui ancaman RAT.

BAB IV HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

Adanya tahap perancangan sistem dilakukan agar peneliti mendapatkan gambaran terhadap bagaimana implementasi serta integrasinya antara satu komponen dalam sistem ataupun jaringan dengan yang lain. Keseluruhan sistem terbagi menjadi dua komponen utama, yaitu pada sisi penyerang serta sisi target pengguna, dengan seluruh layanan dijalankan menggunakan docker container. Pada sisi pengguna, perancangan ditunjukkan untuk mengembangkan aplikasi desktop GUI yang dijadikan sebagai target kerentanan Apache Log4j. **Dalam** kasus ini, target aplikasi berupa program autentikasi lokal sederhana dengan adanya integrasi fitur riwayat menggunakan Apache Log4j. Pada sisi penyerang, perancangan mencakup pengembangan payload RAT, perangkat BadUSB, serta beberapa layanan di dalamnya yang terintegrasi untuk melakukan penyerangan secara utuh. Perancangan sistem meliputi desain topologi jaringan yang digunakan serta struktur skema penyimpanan LDAP untuk sisi penyerang, sedangkan perancangan aplikasi, seperti pembentukan class diagram, akan dimasukkan kedalam sub bab implementasi sistem. Pada tabel 4.1 berikut adalah spesifikasi perangkat dalam merancang dan mengimplementasikan sistem:

Tabel 4.1 Spesifikasi perangkat

No.	Perangkat Keras	Spesifikasi	
1	ASUS VivoBook 14 X407UAR (laptop A)	Processor	Intel Core i3-7020U
		OS	Linux Mint 20.3 (una)
		CPU	2.30 GHz
		RAM	12144240 kB
2	HP EliteBook 2560p (laptop B)	Processor	Intel Core i5-2520M
		OS	Linux Mint 20.3 (una)
		CPU	2.50 GHz
		RAM	10107488 kB

3	DigiSpark Attiny 85 Mini USB Dev. Board	Flash Memory	6 kB + 2kB bootloader
		LED	Power + Status (pin0)
No.	Perangkat Virtual	Spesifikasi	
1	ldap-http-attacker (container A)	OS	Ubuntu Server 20.04
		Shell	/bin/bash
		Port Bindings	2000 — 2100 / tcp
		Network	172.17.0.1 / 16
		IP Address	172.17.0.2
No.	Perangkat Lunak		
1	Apache HTTP Server (2.4.41)		
2	OpenLDAP Server (2.4.49)		
3	Oracle Java SDK (1.8.0_181) & (1.8.0_333)		
4	Apache Maven (3.6.3)		
5	Apache Log4j (2.14.1) , (2.15.0) , (2.16.0) & (2.17.0)		
6	Go (1.18.3)		
7	Arduino IDE (1.8.19)		

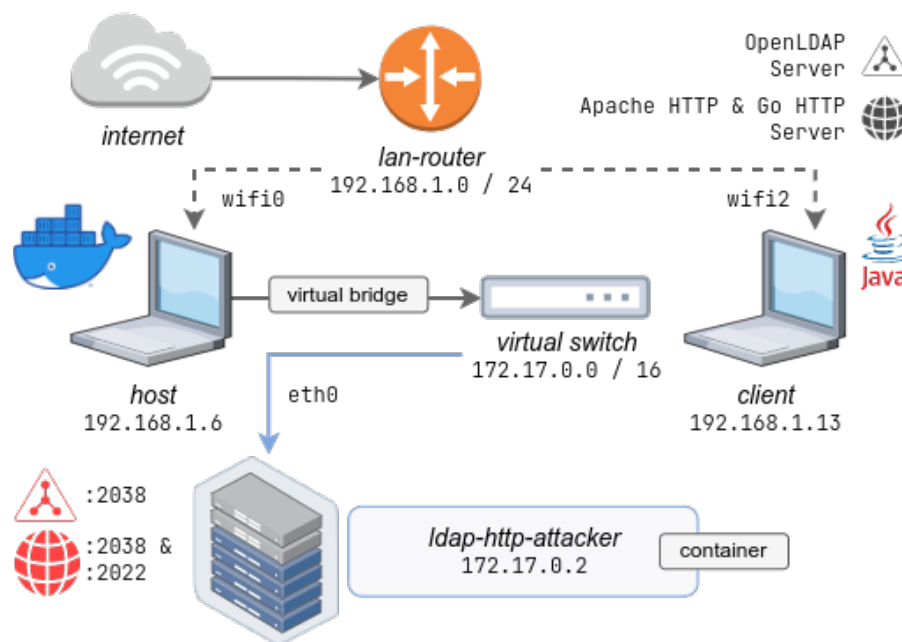
4.1.1 Desain Topologi Jaringan

Dalam membangun keseluruhan sistem, adapun topologi jaringan yang dirancang untuk menggambarkan keseluruhan arsitektur jaringan terhadap setiap komponen di dalamnya. Berikut merupakan keterangan terhadap komponen topologi jaringan yang dipaparkan pada gambar 4.1 yang direferensikan pada tabel 4.1 diatas:

1. Topologi menggunakan dua buah mesin laptop, yaitu laptop A untuk menjalankan berbagai layanan yang dibutuhkan selama proses pengujian serta integrasinya dengan aplikasi, serta laptop B yang didedikasikan sebagai sisi pengguna sebagai target pengujian. Dalam konteks pengujian ini, laptop A juga akan dimanfaatkan sebagai sisi penyerang untuk menjalankan mayoritas dari seluruh tahap penyerangan
2. Dalam laptop A, seluruh layanan sisi penyerang yang dibutuhkan akan dijalankan menggunakan virtualisasi docker container. Pada container A, layanan yang dibangun adalah layanan LDAP menggunakan OpenLDAP

pada nomor port :2038, serta layanan Hypertext Transfer Protocol (HTTP) menggunakan Apache HTTP Server serta aplikasi Go HTTP pada nomor port :2022 dan :2080

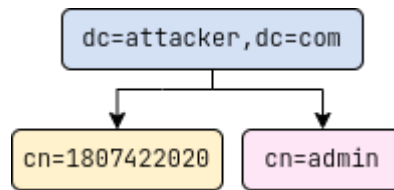
3. Lingkup topologi jaringan berupa skala Local Area Network (LAN)



Gambar 4.1 Topologi jaringan

4.1.2 Desain Skema LDAP

Adanya perancangan desain skema LDAP dapat menjabarkan bentuk struktural penyimpanan suatu entri. Pemodelan skema LDAP pada penelitian ini didasarkan pada struktur Directory Information Tree (DIT). Salah satu komponen di dalamnya adalah Relative Distinguished Name (RDN), yang digunakan untuk mengidentifikasi suatu entri. Pada implementasinya, seluruh susunan RDN dari tingkatan paling dasar hingga menuju RDN entri, merupakan alamat lengkap yang dapat digunakan untuk menavigasikan pencarian entri dalam layanan. Alamat lengkap entri tersebut disebut juga sebagai Distinguished Name (DN) (ZyTrax, 2022). Berikut merupakan pemodelaan DIT serta atribut LDAP pada setiap entri yang digunakan dalam sisi penyerang serta keterangan dari penggunaan atribut tersebut, pada gambar 4.2 dan tabel 4.2:



Gambar 4.2 Skema DIT LDAP pada sisi penyerang

Pada gambar 4.2 diatas, tingkat dasar RDN yang digunakan oleh skema LDAP penyerang adalah `dc=attacker,dc=com`, yang menggambarkan bahwa domain pada layanan ini dirancang khusus untuk penyerangan. Dalam model tersebut, hanya terdapat satu entri yang digunakan untuk menavigasikan layanan LDAP terhadap payload yang tersimpan dalam layanan HTTP penyerang. Atribut RDN yang digunakan oleh entri tersebut adalah Common Name (CN), yang merupakan atribut umum untuk menamakan suatu entri tanpa adanya spesifikasi khusus. Adapun entri admin yang otomatis terbuat oleh sistem untuk melakukan berbagai macam operasi pada pengelolaan skema. Berikut pada tabel 4.2 dibawah merupakan keterangan dari penggunaan atribut entri untuk menyimpan referensi alamat payload dalam layanan yang berbeda, dalam kasus ini yaitu HTTP:

Tabel 4.2 Keterangan atribut skema LDAP pada sisi penyerang

RDN	Atribut	
cn=admin	cn	admin
	description	LDAP administrator
cn=1807422020	cn	1807422020
	javaClassName	http://192.168.1.6:2022/Payload.class
	javaCodebase	http://192.168.1.6:2022/
	javaFactory	Payload

Pada tabel 4.2, untuk dapat menyimpan referensi alamat payload yang akan dibuat, maka object class yang digunakan adalah `javaNamingReference` yang memiliki tiga atribut utamanya. Atribut yang pertama, `javaClassName`, berisikan alamat Uniform Resource Identifier (URI) dari payload yang direferensikan. Sedangkan dua atribut lainnya merupakan komponen dari atribut `javaClassName`,

yaitu javaCodebase, yang berisikan alamat Uniform Resource Locator (URL) dari layanan HTTP untuk mengindekskan payload-nya, serta javaFactory, yang berisikan nama berkas dari payload dengan ekstensi class Java. Dikarenakan object class javaNamingReference merupakan tipe auxiliary, atau sebatas karakteristik tambahan, maka entri membutuhkan object class seperti device yang bertipe struktural. Tidak hanya digunakan sebagai dasar dari object class pada entri, namun object class device hanya membutuhkan satu atribut wajib, yaitu penggunaan CN, sehingga tidak ada ketergantungan dengan penambahan atribut yang tidak dibutuhkan. Berikut merupakan contoh dari alamat DN dalam skema yang dapat terbentuk:

- cn=admin,dc=attacker,dc=com
- cn=1807422020,dc=attacker,dc=com

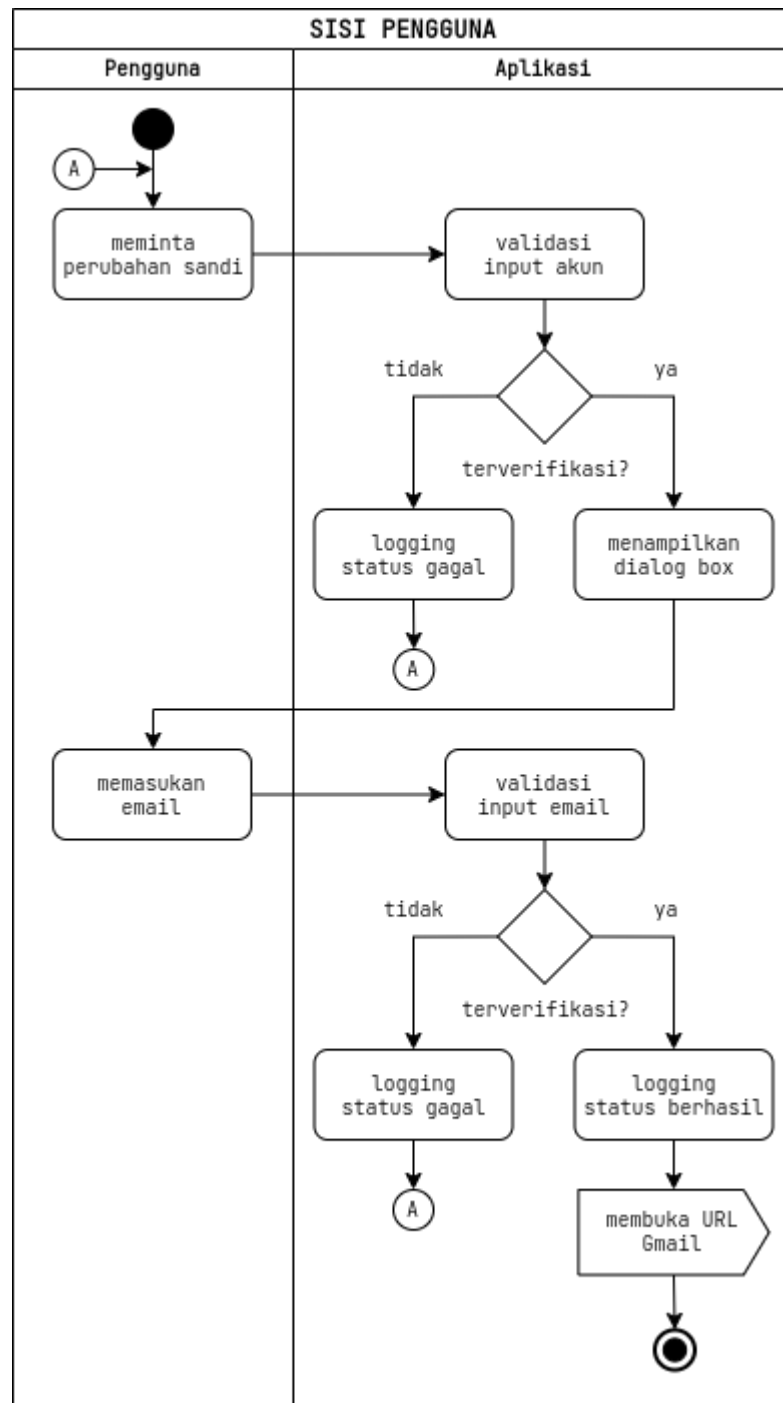
4.2 Implementasi Sistem

Tahap pengimplementasian sistem menjabarkan bagaimana realisasi perancangan terhadap sistem yang akan dibangun. Pembahasan pada bagian ini akan dibagi menjadi dua, yaitu pada sistem pengguna serta penyerang, mulai dari membangun layanan LDAP dan HTTP, aplikasi, serta modul pengujiannya.

4.2.1 Implementasi Sistem Pengguna

4.2.1.1 Pengembangan Aplikasi GUI Desktop

Aplikasi GUI berbasis desktop ini dibangun dalam bahasa Java, agar dapat terintegrasi dengan library Apache Log4j. Dalam rangka penyederhanaan lingkup pengujian, target aplikasi hanya ditunjukkan untuk melakukan fungsi autentikasi berbasiskan kata sandi secara lokal. Adapun data autentikasi, yang mencakup data profil akun, yang disiapkan dalam source code atau secara hardcoded. Aplikasi memiliki dua fitur utama yang akan terhubung dengan fungsi logging dari Apache Log4j, yaitu fitur login serta permintaan untuk perubahan sandi. Hasil dari aksi kedua fitur tersebut akan terekam ke dalam suatu berkas log, baik aksi tersebut berhasil dilakukan atau tidak. Berikut merupakan alur kerja fitur aplikasi yang digambarkan dalam activity diagram pada gambar 4.3 dan 4.4:



Gambar 4.3 Activity diagram pada fitur Request Password Reset aplikasi GUI Desktop

penjelasan activity diagram disini

[act. diag client (user // gui // ldap-client)]

[structure tree]
 [class diagram]
 [pom.xml]
 [snippet properties, Log4j, LDAP Operation, LogPanel]
 [minimum viable product]

4.2.1.1 Instalasi dan Konfigurasi Layanan OpenLDAP

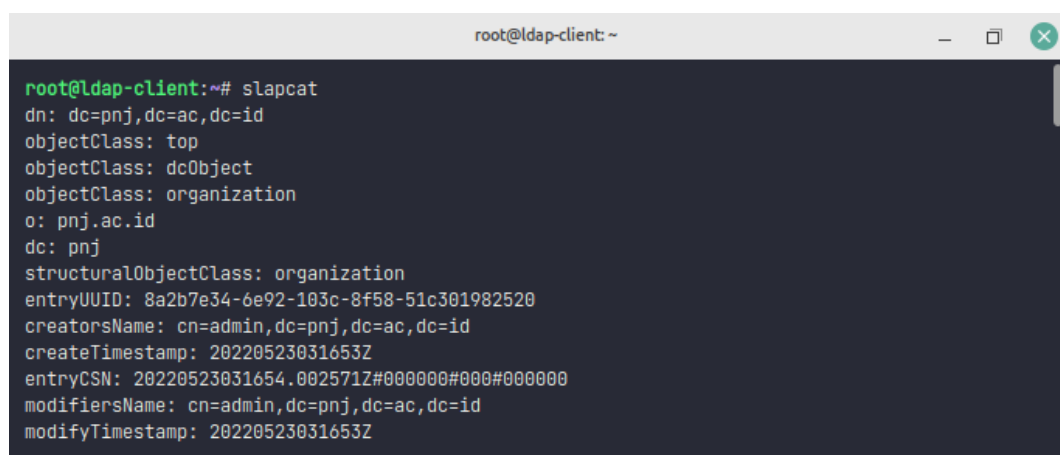
Proses pengembangan layanan LDAP dilakukan di dalam container A, beserta dengan spesifikasi yang dijabarkan pada tabel 4.1. Adapun tahap diawali dengan membuat docker container tersebut lalu melakukan instalasi pada dependensi yang dibutuhkan oleh sistem. Berikut perintah yang digunakan:

```
$ docker pull ubuntu:20.04
$ docker run -p 3000-3100:3000-3100 --hostname "ldap-client" -
it --privileged -e "TERM=xterm-256color" --name "ldap-client"
ubuntu:20.04 /bin/bash
```

Setelah masuk ke dalam shell dari container A, maka instalasi dependensi dapat langsung dilakukan, yang mencakup paket editor teks serta program pendukung untuk layanan OpenLDAP. Selain instalasi, proses juga mencakup konfigurasi layanan seperti pengaturan nomor port serta penyesuaian tingkat dasar RDN dan URI untuk mengakses layanan. Hal ini disesuaikan dengan skema DIT pada gambar 4.2. Perintah yang digunakan sebagai berikut:

```
$ apt-get install net-tools nano curl slapd ldap-utils
$ dpkg-reconfigure slapd
```

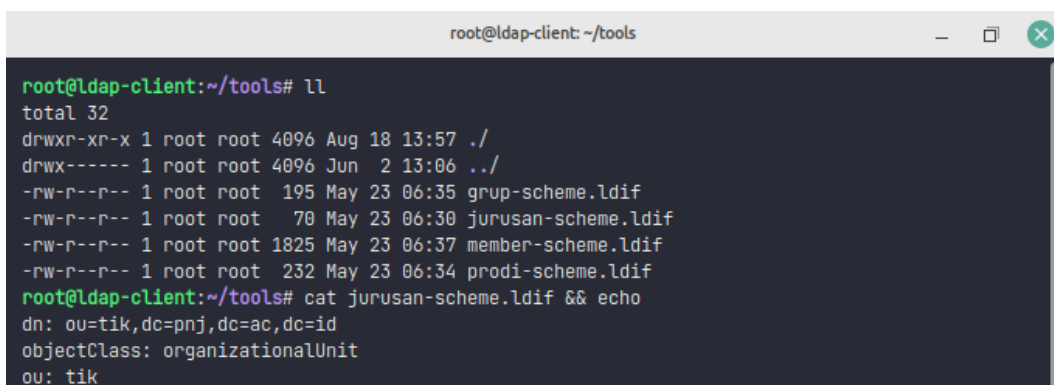
Perintah terakhir akan memberikan menu yang digunakan untuk mengkonfigurasi Domain Name System (DNS) yang merupakan struktur dari tingkat dasar RDN, yaitu pnj.ac.id. DNS tersebut yang nantinya akan dibentuk menjadi Domain Component (DC) yang independen, yaitu dc=pnj,dc=ac,dc=id. Hal ini dapat diverifikasi dengan menggunakan perintah slapcat sebagai berikut:



```
root@ldap-client:~# slapcat
dn: dc=pnj,dc=ac,dc=id
objectClass: top
objectClass: dcObject
objectClass: organization
o: pnj.ac.id
dc: pnj
structuralObjectClass: organization
entryUUID: 8a2b7e34-6e92-103c-8f58-51c301982520
creatorsName: cn=admin,dc=pnj,dc=ac,dc=id
createTimestamp: 20220523031653Z
entryCSN: 20220523031654.002571Z#000000#000#000000
modifiersName: cn=admin,dc=pnj,dc=ac,dc=id
modifyTimestamp: 20220523031653Z
```

Gambar 4.4 Membuat RDN dasar pada layanan LDAP pengguna

Pada gambar 4.4 diatas, dikarenakan RDN tersebut merupakan entri dasar, maka nilai DN juga akan sama dengan RDN nya. Hal ini juga dapat dilihat pada DN dari admin secara lengkap pada atribut creatorsName. Untuk menambahkan entri lainnya, maka akan dimanfaatkan penggunaan berkas LDIF untuk mempermudah pembuatan entri. Atribut entri akan disesuaikan dengan spesifikasi skema LDAP pada rancangan di tabel 4.2. Berikut pada gambar 4.5 adalah isi dari salah satu berkas LDIF, yang seluruhnya tertera pada lampiran 4 hingga 8:



```

root@ldap-client: ~/tools
root@ldap-client:~/tools# ll
total 32
drwxr-xr-x 1 root root 4096 Aug 18 13:57 ./
drwx----- 1 root root 4096 Jun  2 13:06 ../
-rw-r--r-- 1 root root  195 May 23 06:35 grup-scheme.ldif
-rw-r--r-- 1 root root   70 May 23 06:30 jurusan-scheme.ldif
-rw-r--r-- 1 root root 1825 May 23 06:37 member-scheme.ldif
-rw-r--r-- 1 root root  232 May 23 06:34 prodi-scheme.ldif
root@ldap-client:~/tools# cat jurusan-scheme.ldif && echo
dn: ou=tik,dc=pnj,dc=ac,dc=id
objectClass: organizationalUnit
ou: tik

```

Gambar 4.5 Contoh berkas LDIF pada entri jurusan

Untuk menambah entri yang terdapat dalam berkas-berkas tersebut ke dalam layanan LDAP, perintah `ldapadd` dapat digunakan dengan menspesifikasikan DN dari entri admin sebagai peran pengelolanya. Adapun perintah harus dilakukan secara berurutan sesuai dengan tingkatan entri dalam skema pada gambar 4.2 agar seluruh alamat RDN bersifat valid:

```

$ ldapadd -x -D cn=admin,dc=pnj,dc=ac,dc=id -W -f jurusan.ldif
$ ldapadd -x -D cn=admin,dc=pnj,dc=ac,dc=id -W -f prodi.ldif
$ ldapadd -x -D cn=admin,dc=pnj,dc=ac,dc=id -W -f grup.ldif
$ ldapadd -x -D cn=admin,dc=pnj,dc=ac,dc=id -W -f member.ldif

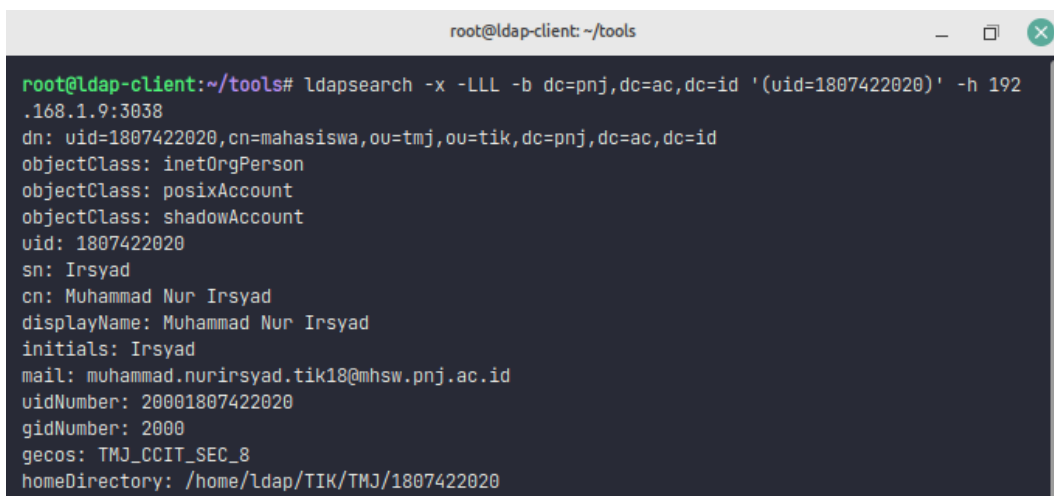
```

Setelah seluruh entri berhasil dimasukkan, maka layanan dapat dilakukan konfigurasi untuk nomor port terlebih dahulu sebelum diverifikasi salah satu entrinya. Merujuk pada topologi jaringan di gambar 4.2, nomor port layanan LDAP akan diubah dari nomor `:389` menjadi `:3089`, yang masih dalam jangka

untuk port binding agar mesin dapat mengaksesnya dari luar container. Berikut merupakan perubahan berkas konfigurasi menggunakan editor teks nano pada `/etc/default/slapd` dan `/etc/ldap/ldap.conf`:

```
$ nano /etc/default/slapd
- SLAPD_SERVICES="ldap://:3038/ ldapi:/// "
$ nano /etc/ldap/ldap.conf
- BASE    dc=pnj,dc=ac,dc=id
- URI     ldap://172.17.0.3:3038
```

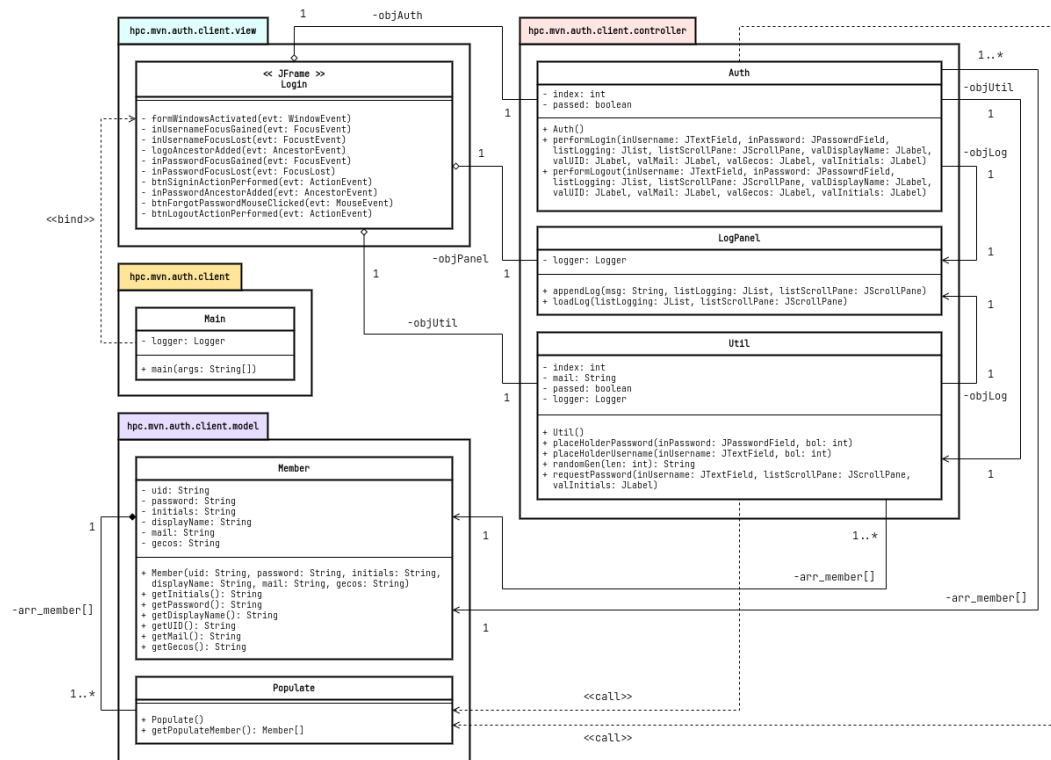
Untuk memverifikasi entri yang sudah tersimpan, perintah yang dapat digunakan adalah `ldapsearch`. Berikut pada gambar 4.6 merupakan contoh pencarian entri dari mahasiswa TMJ dengan menspesifikasikan atribut UID-nya, yang akan diakses menggunakan alamat IP mesin laptop A untuk menguji fungsi port binding dari container A terhadap koneksi luar:



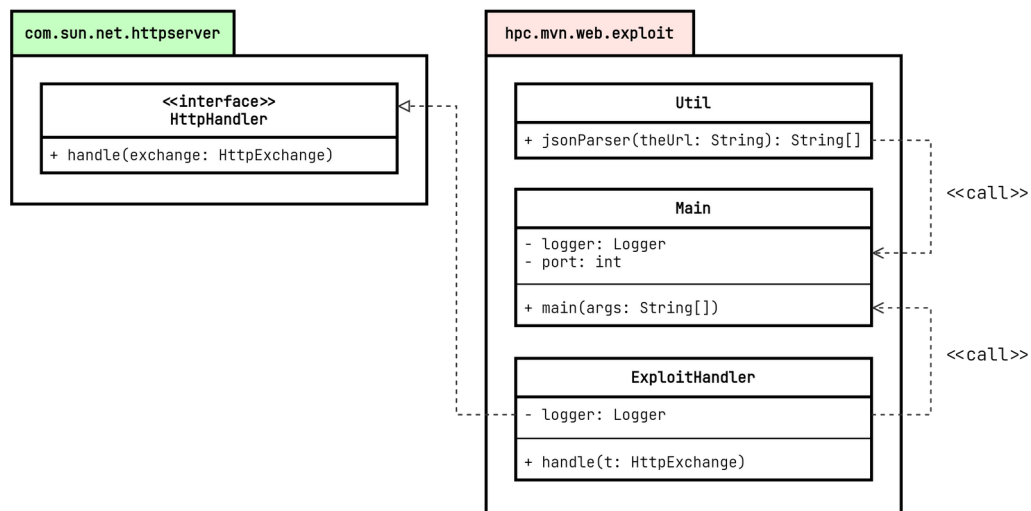
```
root@ldap-client: ~/tools
root@ldap-client:~/tools# ldapsearch -x -LLL -b dc=pnj,dc=ac,dc=id '(uid=1807422020)' -h 192
.168.1.9:3038
dn: uid=1807422020,cn=mahasiswa,ou=tmj,ou=tik,dc=pnj,dc=ac,dc=id
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: 1807422020
sn: Irsyad
cn: Muhammad Nur Irsyad
displayName: Muhammad Nur Irsyad
initials: Irsyad
mail: muhammad.nurirsyad.tik18@mhs.w.pnj.ac.id
uidNumber: 20001807422020
gidNumber: 2000
gecos: TMJ_CCIT_SEC_8
homeDirectory: /home/ldap/TIK/TMJ/1807422020
```

Gambar 4.6 Verifikasi entri mahasiswa dalam layanan LDAP pengguna

4.2.1.2 Pengembangan Aplikasi GUI Desktop LDAP Client



Aa



aa

```
[ act. diag client (user // gui // ldap-client) ]
```

```
[ class diagram ]
```

[pom.xml]

[snippet properties, Log4j, LDAP Operation, LogPanel]

[minimum viable product]

4.2.2 Implementasi Sistem Penyerang

[server ldap serta java http + payload]

4.2.2.1 Instalasi dan Konfigurasi OpenLDAP Server

[spesifikasi docker]

[docker pull + docker exec]

[install open ldap]

[Idif dari skema ldap bab 3]

[test ldapsearch attacker]

4.2.2.2 Instalasi dan Konfigurasi Apache HTTP Server

[pakai container yg sudah ada]

[install apache2]

[proses buat virtual host + touch file payload]

[test curl + lynx]

4.2.2.3 Pengembangan Aplikasi Java HTTP Server

[act. diag attacker (user // http server) & client (system)]

[class diagram]

[pom.xml]

[snippet properties, socketaddr, Log4j, header payload]

[minimum viable product]

4.2.2.4 Pengembangan Aplikasi Go HTTP Server

[act. diag attacker (user // go api) & client (system)]

[route endpoint]

[snippet struct, JSON parser, router]

[minimum viable product]

4.2.2.5 Pengembangan Payload Java

[snippet properties, nama Object, reverseshell]

[minimum viable product]

4.2.2.6 Pengembangan BadUSB

[instalasi + setup full]

[pembuatan base64 script]

4.3 Pengujian Aplikasi dan Sistem

[pengujian whitebox, baik untuk aplikasi dan kerentanan sistem]

4.3.1 Prosedur Pengujian Aplikasi

[uji 1 : integration testing + unit testing]

4.3.1.1 integration Testing

[client]

- LDAP Context
- Log4j Rolling Files
- Config Properties

[attacker java http]

- Remote Config Properties
- Log4j Rolling Files

[attacker java payload]

- Remote Config Properties

[bad USB]

- Open Gnome Program (Calculator)

4.3.1.2 Unit Testing

[client]

- LDAP Authentication Entries
- Log4j Message Lookup Subtitution
- Remote JNDI Lookup Context

[attacker java http]

- Custom HTTP Header Request
- Log4j Message Lookup Subtitution

[attacker java payload]

- Local Encrypted Reverse Shell

[bad USB]

- Local Payload Injection
- Curl Java HTTP Service

[test coverage]

4.3.2 Prosedur Pengujian Kerentanan Sistem

[uji 2 : PTES]

4.3.2.1 Pre-Engagement

[dokumentasi]

4.3.2.2 Intelligence Gathering

[dalemin info info aplikasi gui + sistem client]

[OWASP dependency check]

[OSSIndex Maven]

4.3.2.3 Threat Modelling

[attended : act. diag ☺ client (user // gui) & attacker (ldap // http // system)]

[unattended : act. diag ☺ client (system) & attacker (java // ldap // http // system)]

[aset primer]

[aset sekunder]

4.3.2.4 Vulnerability Analysis

[dalemin cve-2021-44228]

[bikin cvss internal, base score ambil dari official, kita yg environ]

[attack trees]

[deskripsi lab testing]

[hardware spec + container + bad usb]

[software spec + tools]

4.3.2.5 Exploitation

[berdasarkan attack tree : 2 attack vector]

[BadUSB M alware + Hands-on-Keyboards]

4.3.2.6 Post-Exploitation

[cronjob– daemon persistence]

[libprocesshider.c – hide process]

4.3.2.7 Reporting

[mitigasi untuk exploit & post-exploitation]

4.3.2.8 Post-Mitigation Exploitation

[ulang tahapan exploit & post-exploitation]

4.4 Hasil Pengujian Aplikasi dan Sistem

[hasil pengujian whitebox, baik untuk aplikasi dan kerentanan sistem]

4.4.1 Evaluasi Hasil Pengujian Aplikasi

[1 : pengembangan sistem dan tools instrumen penelitian untuk wbox]

4.4.2 Evaluasi Hasil Pengujian Kerentanan Sistem

[2 : tingkat keberhasilan mitigasi terhadap ancaman RAT]

[2 : pengaruh performa sistem terhadap ancaman RAT]

BAB V

PENUTUP

5.1 Kesimpulan

ABC

5.2 Saran

ABC

DAFTAR PUSTAKA

- Achmad, Y.F. and Yulfitri, A. (2020) 'Pengujian Sistem Pendukung Keputusan Menggunakan Black Box Testing Studi Kasus E-Wisudawan Di Institut Sains Dan Teknologi Al-Kamal', *Jurnal Ilmu Komputer*, 5, p. 42.
- Apache (2021) *Apache Log4j Security Vulnerabilities*, Apache Software Foundation. Available at: <https://logging.apache.org/log4j/2.x/security.html> (Accessed: 17 March 2022).
- Apache (2022) *Apache Log4j 2 v. 2.17.2 User's Guide*, Apache Software Foundation. Available at: <https://logging.apache.org/log4j/2.x/log4j-users-guide.pdf> (Accessed: 31 March 2022).
- Biswas, S. *et al.* (2018) *A Study on Remote Code Execution Vulnerability in Web Applications*, *International Conference on Cyber Security and Computer Science*. Available at: <https://www.researchgate.net/publication/328956499>.
- Bojović, P.D. *et al.* (2019) 'The rising threat of hardware attacks: A keyboard attack case study', (November), pp. 1–7. Available at: <https://www.researchgate.net/publication/331312670>.
- Calín, M. *et al.* (2020) *Software Vulnerabilities Overview: A Descriptive Study*, *Tsinghua Science and Technology*. doi:10.26599/TST.2019.9010003.
- CEH (2013) *Trojans and Backdoors - Module 06*, EC-Council. Available at: <http://securitywatch.pcmag.com>.
- Cisco (2021) *Vulnerabilities in Apache Log4j Library Affecting Cisco Products: December 2021*. Available at: <https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd>.
- CVE (2021) *CVE-2021-44228*, CVE Mitre Org. Available at: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228> (Accessed: 4 May 2022).
- Dalalana, D.B. and Zorzo, A.F. (2017) 'Overview and Open Issues on Penetration Test', *Journal of the Brazilian Computer Society*, 23(1). doi:10.1186/s13173-017-0051-1.

FIRST (2019) 'Common Vulnerability Scoring System version 3.1 Specification Document Revision 1', pp. 1–24. Available at: <https://www.first.org/cvss/>.

Hama Saeed, M.A. (2020) 'Malware in Computer Systems: Problems and Solutions', *IJID (International Journal on Informatics for Development)*, 9(1), p. 1. doi:10.14421/ijid.2020.09101.

Helmke, M., Hudson, A. and Hudson, P. (2019) *Ubuntu Unleashed: 2019 Edition*, Pearson Education, Inc.

Hiesgen, R. *et al.* (2022) 'The Race to the Vulnerable: Measuring the Log4j Shell Incident'. Available at: <http://arxiv.org/abs/2205.02544>.

Ingoldsby, T.R. (2021) *Attack Tree-based Threat Risk Analysis*, Amenaza Technologies Limited. Available at: www.amenaza.com.

Ismail, N.M. (2020) 'Rancang Bangun Aplikasi Gamifikasi Untuk Hafalan Al-Quran Menggunakan Audio Fingerprint Berbasis Android'.

Jamil, M.A. *et al.* (2017) 'Software testing techniques: A literature review', *Proceedings - 6th International Conference on Information and Communication Technology for the Muslim World, ICT4M 2016*, pp. 177–182. doi:10.1109/ICT4M.2016.40.

Kaushik, K. *et al.* (2021) 'A Novel Approach to Generate a Reverse Shell: Exploitation and Prevention', *International Journal of Intelligent Communication, Computing, and Networks*, 2(2). doi:10.51735/ijiccn/001/33.

Khan, A. and Neha, R.P. (2016) 'Analysis of Penetration Testing and Vulnerability in Computer Networks', *GRD Journals-Global Research and Development Journal for Engineering* |, 1(6). Available at: www.eeye.com.

LiveAction (2022) *Hands On Keyboard Attack: Why Detection Just Became Critical*. Available at: <https://www.liveaction.com/resources/blog/hands-on-keyboard-attack-why-detection-just-became-critical/#:~:text=A hands-on keyboard attack,other end of this technique>.

Madhavi, D. (2016) 'A White Box Testing Technique in Software Testing: Basis Path Testing', *Journal for Research*, 2(4), pp. 12–17. Available at: www.journalforresearch.org.

- Maraj, A., Rogova, E. and Jakupi, G. (2020) *Testing of Network Security Systems through DoS, SQL Injection, Reverse TCP and Social Engineering Attacks*, *Int. J. Grid and Utility Computing*. doi:10.1504/IJGUC.2020.103976.
- Midian, P. (2002) 'Perspectives on penetration testing - Black box vs. white box', *Network Security*, 2002(11), pp. 10–12. doi:10.1016/S1353-4858(02)11009-9.
- Muñoz, A. and Mirosh, O. (2016) *A Journey from JNDI/LDAP Manipulation to Remote Code Execution Dream Land, BlackHat USA*. Available at: <https://www.blackhat.com/> (Accessed: 14 March 2022).
- Nanny, Prayudi, Y. and Riadi, I. (2019) 'Peningkatan Keamanan Data Terhadap Serangan Remote Access Trojan (RAT) pada Cybercriminal Menggunakan Metode Dynamic Static', *Jurnal Instek*, 4(2), pp. 161–170.
- Ningsih, S.W. (2021) 'Analisis Pengujian Kerentanan Situs Pemerintahan XYZ dengan PTES', *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 8(3), pp. 1543–1556. doi:10.35957/jatisi.v8i3.1224.
- OMG (2011a) *Activity Diagrams*. Available at: <https://www.uml-diagrams.org/activity-diagrams.html>.
- OMG (2011b) *UML Class and Object Diagrams Overview*. Available at: <https://www.uml-diagrams.org/class-diagrams-overview.html>.
- Oracle (2010) *inetOrgPerson Object Class*, *Oracle Corporation*. Available at: <https://docs.oracle.com/cd/E19225-01/820-6551/bzbpb/index.html> (Accessed: 5 May 2022).
- Oracle (2021) *Oracle Security Alert Advisory - CVE-2021-44228*, *Oracle Corporation*. Available at: <https://www.oracle.com/security-alerts/alert-cve-2021-44228.html> (Accessed: 17 March 2022).
- PTES (2021) *The Penetration Testing Execution Standard Documentation - Release 1.1*, *The PTES Team*. Available at: <https://pentest-standard.readthedocs.io/en/latest/tree.html> (Accessed: 3 April 2022).
- Rajasinghe, R. (2022) 'Remote Code Execution Security Flaw in Apache Log4j2', (May). doi:10.13140/RG.2.2.14272.20486.
- Roy, U.K. (2015) *Advanced Java programming*, *Oxford University Press*. Available at: <https://india.oup.com/product/advanced-java-programming-9780199455508> (Accessed: 31 March 2022).

Saroeval, M. and Bhadola, S. (2022) ‘Network Utility Tools Best Practices’, 9(6), pp. 96–103.

Shevchenko, N. *et al.* (2018) *Threat Modeling: A Summary Of Available Methods*, Carnegie Mellon University: Software Engineering.

Sukic, C. and Saracevic, M. (2012) ‘UML and JAVA as effective tools for implementing algorithms in computer graphics’, *Tem Journal*, 1(2), p. 111.

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

ZyTrax (2022) *LDAP for Rocket Scientists*, ZyTrax Inc. Available at: <https://www.zytrax.com/books/ldap/> (Accessed: 16 May 2022).