



**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.....	3
LEMBAR PENGESAHAN.....	4
KATA PENGANTAR.....	5
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI.....	6
ABSTRAK.....	7
DAFTAR ISI.....	8
DAFTAR GAMBAR.....	10
DAFTAR TABEL.....	11
DAFTAR ISTILAH.....	12
BAB I PENDAHULUAN.....	13
1.1 Latar Belakang.....	13
1.2 Rumusan Masalah.....	15
1.3 Batasan Masalah.....	15
1.4 Tujuan dan Manfaat.....	16
1.5 Sistematika Penulisan.....	17
BAB II TINJAUAN PUSTAKA.....	18
2.1 Remote Access Trojan.....	18
2.1.1 Reverse & Bind Shell TCP.....	18
2.2 Apache Log4j.....	19
2.2.1 Lightweight Directory Access Protocol.....	20
2.2.2 Kerentanan CVE-2021-44228.....	21
2.3 Penetration Testing Execution Standard.....	21
2.3.1 White Box Testing.....	21
2.3.2 Common Vulnerability Scoring System.....	21
2.3.3 Attack Trees.....	21
2.3.4 Hands-on-Keyboards Attack.....	21
2.3.5 BadUSB Malware.....	21
2.4 Unified Modelling Language.....	21
2.4.1 Class Diagram.....	21
2.4.2 Activity Diagram.....	21
2.5 Software Testing.....	21
2.5.1 Integration Testing.....	21
2.5.2 Alpha Testing.....	22
2.6 Docker Container.....	22
2.7 Penelitian Sejenis.....	22
BAB III METODE PENELITIAN.....	23
3.1 Rancangan Penelitian.....	23
3.2 Tahapan Penelitian.....	23
3.3 Objek Penelitian.....	23
BAB IV HASIL DAN PEMBAHASAN.....	24
4.1 Perancangan Sistem.....	24

4.1.1 Desain Topologi Jaringan.....	24
4.1.2 Desain Skema LDAP.....	24
4.1.3 Desain Class Diagram Aplikasi.....	24
4.2 Implementasi Sistem.....	24
4.2.1 Implementasi Sistem Pengguna.....	24
4.2.1.1 Instalasi dan Konfigurasi OpenLDAP Server.....	24
4.2.1.2 Pengembangan Aplikasi GUI Desktop LDAP Client.....	24
4.2.2 Implementasi Sistem Penyerang.....	24
4.2.2.1 Instalasi dan Konfigurasi OpenLDAP Server.....	24
4.2.2.2 Instalasi dan Konfigurasi Apache HTTP Server.....	25
4.2.2.3 Pengembangan Aplikasi Java HTTP Server.....	25
4.2.2.4 Pengembangan Aplikasi Go JSON REST API.....	25
4.2.2.5 Pengembangan Payload Java.....	25
4.2.2.6 Pengembangan BadUSB.....	25
4.3 Pengujian Aplikasi dan Sistem.....	25
4.3.1 Prosedur Pengujian Aplikasi.....	25
4.3.1.1 integration Testing.....	26
4.3.1.2 Alpha Testing.....	26
4.3.2 Prosedur Pengujian Kerentanan Sistem.....	26
4.3.2.1 Pre-Engagement.....	27
4.3.2.2 Intelligence Gathering.....	27
4.3.2.3 Threat Modelling.....	27
4.3.2.4 Vulnerability Analysis.....	27
4.3.2.5 Exploitation.....	27
4.3.2.6 Post-Exploitation.....	27
4.3.2.7 Reporting.....	27
4.3.2.8 Post-Mitigation Exploitation.....	27
4.4 Hasil Pengujian Aplikasi dan Sistem.....	27
4.4.1 Evaluasi Hasil Pengujian Aplikasi.....	27
4.4.2 Evaluasi Hasil Pengujian Kerentanan Sistem.....	27
BAB V PENUTUP.....	28
5.1 Kesimpulan.....	28
5.2 Saran.....	28
DAFTAR pustaka.....	29

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 URL Entry Manipulation. Dua bentuk attack vector 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 penelitian dan integrasinya dengan Apache Log4j yang sesuai dengan referensi CVE-2021-44228?
2. Bagaimana pengujian serta mitigasi terhadap tingkat keberhasilannya pada kerentanan Apache Log4j, terkait ancaman Remote Access Trojan dalam lingkup white box testing?
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 penelitian
 - 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
 - c) Mesin komputer yang dipakai berbasis platform Linux, sehingga seluruh payload, program, serta skrip akan disesuaikan ke arah tersebut
2. Batasan dalam implementasi pengujian dan mitigasinya
 - a) Pengujian berbasis metode PTES dengan pendekatan white box testing yang hanya ditujukan untuk library Apache Log4j dalam versi 2.0-beta9 hingga 2.14.1 (referensi CVE-2021-44228), sehingga tidak mencakup *security patch* dalam versi setelahnya, yaitu versi 2.15.0 (26/11/2021), ataupun versi yang terakhir yaitu 2.17.1 (28/12/2021)

- b) Bentuk mitigasi mencakup pendekatan deteksi ancaman, serta implementasi *static analysis* dan pemanfaatan program pemantauan dan konfigurasi internal sistem
 - 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
- 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 penelitian, 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

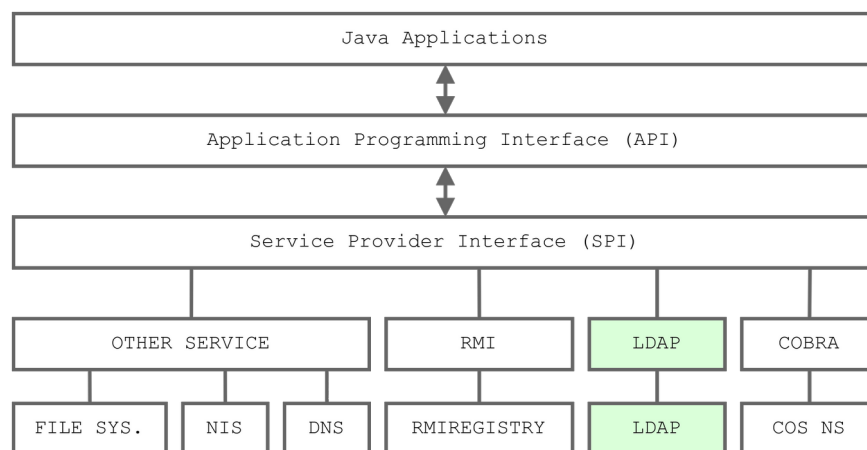
Bind shell bekerja dengan membuka layanan koneksi TCP di mesin korban pada port tertentu, yang juga disebut sebagai listener. Koneksi tersebut kemudian disambungkan oleh mesin penyerang untuk mendapatkan shell korban melalui remote access nya. Dikarenakan listener dilakukan dari mesin korban, hal ini 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 yang relevan, serta melakukan suatu pemetaan terhadap aset primer dan sekunder yang dapat ditargetkan
4. Vulnerability Analysis: menganalisis celah kerentanan untuk mendapatkan serangan yang efektif, serta mendefinisikan jalur serangan dan lingkungan pengujian untuk dilakukan pada tahap selanjutnya
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 dalam pendekatan mitigasi (Ningsih, Almaarif and Widjajarto, 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 publikasi, grup Base dapat menggambarkan nilai kerentanan secara umum pada periode yang lama untuk diaplikasikan ke dalam berbagai lingkungan pengujian (FIRST, 2019). Pada tabel 2.2 berikut merupakan deskripsi dan parameter dari metrik pada grup Base dalam CVSS versi 3.1:

Tabel 2.2 Keterangan metrik grup Base pada CVSS versi 3.1





Metrik	Deskripsi	Parameter	
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

2.4.2 Attack Trees

Attack tree merupakan framework untuk menggambarkan bagaimana rangkaian pengujian dapat dilakukan, berdasarkan tujuan utama yang disusun dengan struktur pohon. Attack tree didasarkan pada perspektif bagaimana penyerang melakukan eksploitasi, dan bagaimana meraih tujuan utamanya. Attack tree dapat memiliki beberapa sub tujuan, atau node, di dalam tujuan utamanya, yang disebut juga sebagai root node. Dalam tingkatannya, setiap node memiliki leaf node untuk menggambarkan serangan yang dibutuhkan dalam meraih tujuan node tersebut, untuk melanjutkan ke node di atasnya. Setiap node bersifat AND atau OR, yang digunakan untuk mendeskripsikan syarat suksesi dari tujuan node terhadap node dan leaf node 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-Keyboards

Serangan hands-on-keyboard merupakan pendekatan yang terjadi di mana penyerang sudah berada di dalam lingkungan sistem target, lalu menggunakan 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, penggunaan deteksi keystroke 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 dapat mengemulasi Human Interface Device (HID) dalam sistem target, seperti keyboard, mouse, touchpad, hingga pemindai sidik jari. Dikarenakan penggunaan perangkat HID tidak dilakukan pemindaian, 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 kemudian terbantu dengan kecepatan keystroke per huruf hingga milidetik untuk menyelesaikan seluruh injeksinya, sehingga durasi serangan dapat berkurang secara signifikan (Bojović *et al.*, 2019).

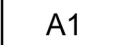
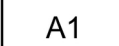
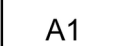
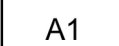


2.5 Unified Modelling Language

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 dan aktivitas penggunaan aplikasi dari berbagai sudut pandang tertentu. 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 Java, yang sama sama berfokuskan pada objeknya, atau object-

oriented, sehingga class diagram juga akan 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:


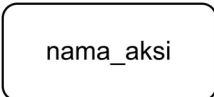

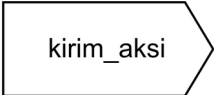
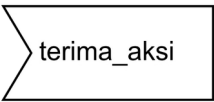
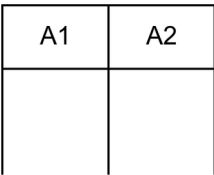
Tabel 2.4 Deskripsi simbol class diagram

Simbol	Nama	Deskripsi
<div> <div>nama_class</div> <div>- atribut: tipe_data</div> <div>+ operasi(tipe_data): tipe_data</div> </div>	Class	klasifikasi karakteristik objek
	Atribut	properti variabel dalam class
	Operasi	fungsi metode dalam class
<div> <div>A1  A2</div> <div>A1  A2</div> <div>A1  A2</div> </div>	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
<div>A1  A2</div>	Dependensi	adanya relasi abstrak terhadap referensi suatu elemen dalam class lain pada lingkup fungsi
<div> <div>  <div>nama_operasi metode / URI</div>  </div> <div>atribut: tipe_data</div> </div>	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. Tujuan utama penggunaan activity diagram yaitu untuk 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, serta 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 relevan terhadap studi kasus untuk mengembangkan aspek analisis. Adapun berikut adalah judul penelitian yang akan dipaparkan sebagai perbandingan penelitian terdahulu yang sejenis dengan topik dan judul penelitian ini:

1. Remote Code Execution Security Flaw in Apache Log4j2

Penelitian tersebut menganalisis eksploitasi kerentanan Apache Log4j pada CVE-2021-44228 serta CVE-2021-4506 terhadap serangan Remote Code Execution (RCE) dalam lingkup white box testing. Peneliti juga mengimplementasikan adanya deteksi serta mitigasi berbasis pada static analysis. Vektor serangan yang digunakan adalah JNDI Injection melalui HTTP header X-API-Version untuk mendapatkan reverse shell TCP menggunakan netcat [sumber]

2. Analisis Pengujian Kerentanan Situs Pemerintahan XYZ dengan PTES

Penelitian tersebut menganalisis pengujian kerentanan yang berbasis metode PTES terhadap aplikasi website lembaga lokal dalam lingkup

black box testing. Peneliti memberikan rekomendasi terhadap mitigasi melalui penggunaan Web Application Firewall (WAF) dan perbaikan konfigurasi baik pada opsi header aplikasi ataupun filterisasi input pengguna. Berdasarkan kerentanan yang ditemukan, pengujian menggunakan vektor serangan Cross Site Scripting (XSS) serta Clickjacking [sumber]

3. Peningkatan Keamanan Data Terhadap Serangan Remote Access Trojan (RAT) pada Cybercriminal Menggunakan Metode Dynamic Static a [sumber]
4. Analisis Kerentanan Log4Shell pada CVE-2021-44228 terhadap Ancaman Remote Access Trojan dengan Metode Penetration Testing Execution Standard

Penelitian ini mengadaptasi berbagai macam aspek dari penelitian sebelumnya sebagai bentuk pengembangan. Penelitian ini menganalisis kerentanan Apache Log4j yang dikhususkan pada CVE-2021-44228 terhadap serangan RCE beserta pengembangnya pada RAT secara white box testing. Selain menggunakan static analysis, penelitian ini menggunakan program pemantauan dan konfigurasi internal sistem dalam mendeteksi dan memitigasi ancaman. Vektor serangan yang digunakan adalah JNDI Injection melalui hands-on-keyboard dan HTTP header User-Agent

BAB III

METODE PENELITIAN

3.1 Rancangan Penelitian

[pendekatan penelitian : kuantitatif → variabel terkontrol]

[jenis penelitian : eksperimen → pre eksperimen (one group pretest posttest)]

[jenis data : sekunder (heaton)]

[teknik pengambilan data : penelitian kepustakaan terdahulu dan studi dokumentasi primer dan sekunder]

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

[analisa 2 : tingkat keberhasilan mitigasi terhadap ancaman RAT]

[analisa 2 : pengaruh performa sistem terhadap ancaman RAT]

[analisa 3 : bagaimana perbedaan 2.14.1 dengan 2.15 ?]

3.2 Tahapan Penelitian

[software testing untuk pendukung whitebox]

[berdasarkan metode PTES secara whitebox]

3.3 Objek Penelitian

[kerentanan apache log4j – cve-2021-44228 pada ancaman RAT]

BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

[rangkum deskripsi – singgung topologi, skema ldap, dan class diagram C vs A]

4.1.1 Desain Topologi Jaringan

[desain topologi]

4.1.2 Desain Skema LDAP

[desain skema ldap C vs A]

4.1.3 Desain Class Diagram Aplikasi

[desain class diagram aplikasi C vs A]

4.2 Implementasi Sistem

[instalasi konfigurasi, serta pembangunan aplikasi dan tools, sisi C vs A]

4.2.1 Implementasi Sistem Pengguna

[server ldap serta aplikasi ldap client berbentuk gui]

4.2.1.1 Instalasi dan Konfigurasi OpenLDAP Server

[docker pull + docker exec]

[install open ldap]

[Idif dari skema ldap bab 3]

[test ldapsearch client]

4.2.1.2 Pengembangan Aplikasi GUI Desktop LDAP Client

[structure tree]

[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

[docker pull + docker exec]

[install open ldap]

[ldif 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

[structure tree]

[class diagram]

[pom.xml]

[snippet properties, socketaddr, Log4j, header payload]

[minimum viable product]

4.2.2.4 Pengembangan Aplikasi Go JSON REST API

[structure tree]

[modul dependency]

[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 Substitution
- Remote JNDI Lookup Context

[attacker java http]

- Custom HTTP Header Request
- Log4j Message Lookup Substitution

[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]

[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., Almaarif, A. and Widjarto, A. (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.