Analisis Perbandingan Keamanan Protokol Telnet dan SSH Menggunakan Wireshark pada Ubuntu Server 26

Aprilla Addivi Tiar Almashinta[1*], Cahyani Adzania Rani Miftahul Jannah[2], Nur Aini Faza[3], Salwa Nurussyifa[4], Vanessa Adelita Nareswari[5], Ridwan Dwi Irawan[6]

[1*]Sistem Informasi
Universitas Duta Bangsa Surakarta
[1*]240101065@mhs.udb.ac.id

[2]Sistem Informasi
Universitas Duta Bangsa Surakarta
[2]240101066@mhs.udb.ac.id

[3]Sistem Informasi
Universitas Duta Bangsa Surakarta
[3]240101080@mhs.udb.ac.id

[4]Sistem Informasi
Universitas Duta Bangsa Surakarta
[4]240101094@mhs.udb.ac.id

[5]Sistem Informasi
Universitas Duta Bangsa Surakarta
[5]240101084@mhs.udb.ac.id

[6]Sistem Informasi
Universitas Duta Bangsa Surakarta
[6]ridwan_dwiirawan@udb.ac.id

Abstrak— **Keamanan komunikasi data menjadi aspek penting dalam administrasi server jarak jauh. Telnet dan Secure Shell (SSH) merupakan dua protokol yang umum digunakan untuk keperluan tersebut, namun memiliki karakteristik keamanan yang sangat berbeda. Penelitian ini bertujuan menganalisis dan membandingkan tingkat keamanan protokol Telnet dan SSH melalui pendekatan eksperimental menggunakan Ubuntu Server pada lingkungan virtual VirtualBox, PuTTY sebagai aplikasi client, serta Wireshark sebagai alat analisis lalu lintas jaringan. Pengujian dilakukan dengan menghubungkan client ke server menggunakan kedua protokol pada port default masing-masing (23 untuk Telnet dan 22 untuk SSH), kemudian lalu lintas data ditangkap dan dianalisis menggunakan filter protokol pada Wireshark. Hasil penelitian menunjukkan bahwa komunikasi Telnet dikirim dalam bentuk plain text sehingga kredensial pengguna dan perintah yang dijalankan dapat terbaca langsung dari hasil tangkapan paket, sedangkan komunikasi SSH menampilkan proses negosiasi algoritma dan key exchange yang diikuti paket data terenkripsi sehingga isi komunikasi tidak dapat dibaca oleh pihak yang melakukan penyadapan. Dapat disimpulkan bahwa SSH memberikan tingkat keamanan yang jauh lebih baik dibandingkan Telnet dan lebih layak digunakan sebagai standar protokol akses jarak jauh pada administrasi server modern, sementara Telnet hanya relevan digunakan pada lingkungan pembelajaran atau simulasi laboratorium yang terisolasi.**

Kata kunci— Telnet, SSH, Keamanan Jaringan, Wireshark, Enkripsi

Abstract— **Data communication security is a critical aspect of remote server administration. Telnet and Secure Shell (SSH) are two protocols commonly used for this purpose, yet they differ significantly in their security characteristics. This study aims to analyze and compare the security level of the Telnet and SSH protocols through an experimental approach using an Ubuntu Server in a VirtualBox virtual environment, PuTTY as the client application, and Wireshark as the network traffic analysis tool. Testing was conducted by connecting the client to the server using both protocols on their respective default ports (23 for Telnet and 22 for SSH), after which the resulting traffic was captured and analyzed using protocol filters in Wireshark. The results show that Telnet communication is transmitted in plain text, allowing user credentials and executed commands to be read directly from the captured packets, whereas SSH communication exhibits an algorithm negotiation and key-exchange process followed by encrypted data packets, preventing the communication content from being read by an eavesdropper. It can be concluded that SSH provides substantially better security than Telnet and is more suitable as the standard remote-access protocol for modern server administration, while Telnet remains relevant only for isolated learning or laboratory simulation environments.**

Keywords— *Telnet, SSH, Network Security, Wireshark, Encryption*

I. PENDAHULUAN

Perkembangan teknologi informasi mendorong meningkatnya kebutuhan pengelolaan server secara jarak jauh (remote administration). Administrator jaringan memerlukan protokol yang mampu memberikan akses ke server secara aman dan efisien tanpa mengorbankan kerahasiaan data yang dipertukarkan antara client dan server [1]. Keamanan jaringan sendiri merupakan upaya untuk melindungi kerahasiaan, integritas, dan ketersediaan informasi dari ancaman pihak yang tidak berwenang, sehingga pemilihan protokol komunikasi yang tepat menjadi salah satu faktor penentu keamanan suatu sistem jaringan [4]. Ancaman terhadap kerahasiaan data pada umumnya

dapat berupa serangan pasif seperti sniffing atau eavesdropping, maupun serangan aktif seperti session hijacking dan Man-in-the-Middle (MITM), yang keduanya dapat dimanfaatkan oleh pihak yang tidak berwenang apabila protokol komunikasi yang digunakan tidak menerapkan mekanisme perlindungan data yang memadai [4].

Dua protokol yang secara historis digunakan untuk keperluan akses jarak jauh adalah Telnet dan Secure Shell (SSH). Telnet (Telecommunication Network) merupakan salah satu protokol remote login tertua yang beroperasi pada port 23 sebagaimana didefinisikan dalam RFC 854, dan memungkinkan pengguna menjalankan perintah pada server dari jarak jauh seolah-olah sedang mengoperasikannya secara langsung [5]. Pada masa awal perkembangan jaringan komputer, Telnet banyak digunakan untuk kebutuhan administrasi server karena proses konfigurasinya yang relatif sederhana serta tidak membutuhkan sumber daya sistem yang besar. Meskipun demikian, Telnet tidak menerapkan mekanisme enkripsi apa pun sehingga seluruh data, termasuk username, password, dan perintah yang dijalankan, dikirimkan dalam bentuk plain text yang rentan terhadap penyadapan (sniffing) oleh pihak yang tidak berwenang [1].

SSH kemudian dikembangkan sebagai solusi atas kelemahan tersebut. Sebagaimana dijelaskan dalam arsitektur protokolnya pada RFC 4251, SSH beroperasi pada port 22 dan menerapkan enkripsi end-to-end pada seluruh sesi komunikasi, termasuk proses autentikasi, sehingga menjadi standar de facto untuk akses jarak jauh pada lingkungan jaringan modern [6]. Arsitektur SSH terdiri atas tiga komponen utama, yaitu Transport Layer Protocol yang menyediakan autentikasi server, kerahasiaan, dan integritas data; User Authentication Protocol yang mengautentikasi client kepada server; serta Connection Protocol yang memultipleks saluran terenkripsi menjadi beberapa kanal logis [6]. Implementasi SSH secara praktis, termasuk mekanisme key exchange dan manajemen kunci publik-privat, dibahas secara rinci sebagai panduan teknis bagi administrator sistem dalam mengamankan layanan akses jarak jauh [7].

Selain pemilihan protokol, keberhasilan pengujian keamanan jaringan turut ditentukan oleh ketersediaan perangkat pendukung yang tepat. Oracle VirtualBox digunakan sebagai perangkat lunak virtualisasi yang memungkinkan sistem operasi server dijalankan secara terisolasi tanpa memengaruhi sistem host, sehingga pengujian dapat dilakukan pada lingkungan yang aman dan terkendali. Ubuntu Server dipilih sebagai sistem operasi server karena bersifat open source, stabil, memiliki dokumentasi yang lengkap, serta banyak digunakan pada lingkungan industri maupun pendidikan. PuTTY digunakan sebagai aplikasi client yang mendukung koneksi baik melalui protokol SSH maupun Telnet, sedangkan Wireshark digunakan sebagai network protocol analyzer untuk menangkap dan menganalisis paket data yang melintasi jaringan sehingga karakteristik keamanan setiap protokol dapat diamati secara langsung pada level paket.

Penelitian terdahulu telah membuktikan secara empiris perbedaan tingkat keamanan kedua protokol tersebut menggunakan analisis lalu lintas jaringan. Hasil simulasi perbandingan Telnet dan SSH pada jaringan terbuka menunjukkan bahwa Telnet mengirimkan data dalam bentuk plaintext yang rentan disadap, sedangkan SSH mengenkripsi seluruh data sehingga jauh lebih aman untuk komunikasi jarak jauh [1]. Di sisi lain, ancaman terhadap layanan SSH terus berkembang, misalnya dalam bentuk serangan brute-force terhadap kredensial login, yang mendorong perlunya mekanisme deteksi berbasis log jaringan [2]. Pendekatan analisis trafik menggunakan Wireshark juga telah diterapkan pada protokol keamanan lain seperti SSL/TLS untuk mengevaluasi tingkat kerahasiaan data pengguna pada layanan berbasis web, dan menunjukkan bahwa metode packet capture efektif digunakan untuk membuktikan secara objektif ada atau tidaknya proses enkripsi pada suatu protokol

komunikasi [3]. Berbagai penelitian lain juga menunjukkan bahwa SSH tanpa pengamanan tambahan tetap berisiko terhadap serangan brute-force maupun key scanning, sehingga administrator disarankan menerapkan mekanisme mitigasi tambahan seperti autentikasi dua langkah maupun port knocking [9].

Berdasarkan uraian tersebut, penelitian ini dilakukan untuk membuktikan secara empiris perbedaan karakteristik keamanan antara Telnet dan SSH melalui pendekatan eksperimental pada lingkungan Ubuntu Server yang dijalankan secara virtual. Rumusan masalah dalam penelitian ini meliputi: (1) bagaimana proses instalasi dan konfigurasi layanan Telnet dan SSH pada Ubuntu Server; (2) bagaimana proses pengujian koneksi menggunakan kedua protokol dilakukan pada lingkungan jaringan virtual; (3) bagaimana karakteristik lalu lintas jaringan yang dihasilkan oleh kedua protokol berdasarkan hasil pengamatan Wireshark; serta (4) mengapa SSH lebih direkomendasikan dibandingkan Telnet untuk administrasi server pada lingkungan jaringan modern. Adapun tujuan penelitian ini adalah mengimplementasikan layanan Telnet dan SSH pada Ubuntu Server, melakukan pengujian koneksi antara client dan server menggunakan kedua protokol, menganalisis lalu lintas jaringan yang dihasilkan oleh kedua protokol menggunakan Wireshark, serta membandingkan tingkat keamanan keduanya berdasarkan hasil implementasi dan pengamatan yang telah dilakukan. Hasil penelitian ini diharapkan dapat menjadi referensi praktis bagi mahasiswa maupun praktisi jaringan mengenai pentingnya migrasi dari protokol lama yang tidak aman menuju protokol dengan mekanisme enkripsi yang lebih baik.

II. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode eksperimen dengan pendekatan deskriptif-komparatif mengikuti kaidah pengujian sistem jaringan pada umumnya, yaitu tahap perancangan topologi, implementasi layanan, pengujian konektivitas, hingga analisis hasil [8]. Implementasi dilakukan secara langsung pada sebuah server, kemudian karakteristik keamanan kedua protokol dibandingkan berdasarkan hasil pengamatan lalu lintas jaringan. Pendekatan eksperimental dipilih karena memungkinkan pembuktian secara langsung dan terukur mengenai ada atau tidaknya mekanisme enkripsi pada suatu protokol, alih-alih hanya mengandalkan kajian pustaka semata.

A. Alat dan Bahan

Perangkat keras yang digunakan berupa satu unit laptop/PC dengan RAM minimal 8 GB sebagai host utama untuk menjalankan mesin virtual. Perangkat lunak yang digunakan meliputi Oracle VirtualBox sebagai aplikasi virtualisasi, Ubuntu Server sebagai sistem operasi server, PuTTY sebagai aplikasi client untuk koneksi SSH maupun Telnet, Wireshark sebagai alat analisis lalu lintas jaringan, serta paket layanan OpenSSH Server dan Telnet Server (xinetd/telnetd) pada sisi server. Seluruh perangkat lunak yang digunakan bersifat open source atau tersedia secara gratis, sehingga metode penelitian ini relatif mudah direplikasi oleh peneliti lain dengan spesifikasi perangkat keras yang setara.

B. Topologi Jaringan

Topologi jaringan yang digunakan bersifat sederhana, terdiri atas satu komputer host dan satu virtual machine Ubuntu Server yang saling terhubung melalui jaringan Host-Only Adapter, sehingga komunikasi antara client dan server dapat diisolasi dari jaringan eksternal selama proses pengujian berlangsung. Penggunaan Host-Only Adapter dipilih agar seluruh lalu lintas data yang ditangkap oleh Wireshark benar-benar berasal dari komunikasi antara client dan server yang diuji, tanpa tercampur oleh trafik jaringan lain yang tidak relevan dengan pengujian.

C. Prosedur Penelitian

Prosedur penelitian dilaksanakan melalui tahapan berikut: (1) instalasi Ubuntu Server pada VirtualBox; (2) konfigurasi jaringan Host-Only Adapter agar komputer host dan

server dapat saling berkomunikasi; (3) verifikasi konektivitas menggunakan perintah ping dari host menuju alamat IP server; (4) instalasi dan aktivasi layanan SSH menggunakan paket OpenSSH Server pada port 22, kemudian diverifikasi statusnya menggunakan perintah systemctl status ssh; (5) instalasi dan konfigurasi layanan Telnet menggunakan xinetd dan telnetd pada port 23, kemudian diverifikasi statusnya menggunakan perintah systemctl status xinetd serta ss -tulnp; (6) pengujian koneksi dari komputer host ke server menggunakan aplikasi PuTTY pada kedua protokol; dan (7) pengambilan (capture) paket data menggunakan Wireshark selama masing-masing sesi koneksi berlangsung.

D. Skenario Pengujian

Skenario pengujian dilakukan dengan login ke server menggunakan SSH pada port 22, menjalankan beberapa perintah dasar (whoami, pwd, hostname, ls) pada terminal, serta melakukan capture paket menggunakan Wireshark selama sesi berlangsung. Langkah yang sama kemudian diulang menggunakan protokol Telnet pada port 23 dengan perintah identik, sehingga hasil tangkapan paket kedua protokol dapat dibandingkan secara objektif pada kondisi pengujian yang setara. Analisis paket pada Wireshark dilakukan menggunakan filter protokol (ssh, telnet) maupun filter port (tcp.port == 22 dan tcp.port == 23) untuk memisahkan trafik yang relevan dari trafik jaringan lainnya.

E. Teknik Pengumpulan dan Analisis Data

Data yang dikumpulkan berupa hasil tangkapan paket (packet capture) dalam format pcapng untuk masing-masing sesi Telnet dan SSH, serta hasil pengecekan status layanan pada terminal Ubuntu Server. Analisis data dilakukan secara deskriptif dengan membandingkan indikator keamanan pada kedua jenis trafik, meliputi keberadaan proses negosiasi enkripsi, keterbacaan payload data pada Follow TCP Stream, serta jenis paket yang mendominasi masing-masing sesi. Hasil analisis kemudian disusun dalam bentuk tabel perbandingan untuk memudahkan interpretasi terhadap tingkat keamanan kedua protokol.

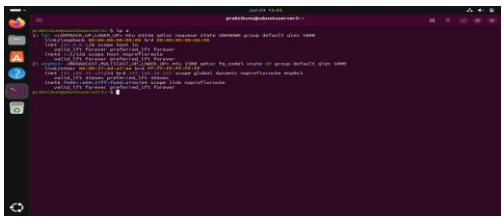
III. HASIL DAN PEMBAHASAN

A. Implementasi Layanan pada Ubuntu Server

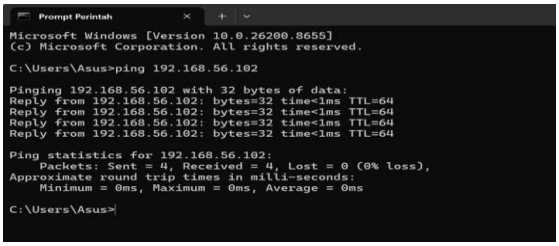
Instalasi Ubuntu Server pada VirtualBox berhasil dilakukan dan sistem operasi dapat berjalan dengan baik pada lingkungan virtualisasi. Proses instalasi meliputi pembuatan virtual machine dengan alokasi memori dasar sebesar 2048 MB, media penyimpanan sebesar 25 GB, serta pengaturan jaringan menggunakan Host-Only Adapter. Setelah proses instalasi selesai, sistem menampilkan layar selamat datang yang menandakan Ubuntu Server telah siap digunakan sebagai media implementasi layanan SSH dan Telnet, sebagaimana ditunjukkan pada Gambar 1.

Gambar 1. Tampilan mesin virtual pada VirtualBox Manager dan layar instalasi Ubuntu Server.

Pengecekan alamat IP menggunakan perintah ip a menunjukkan bahwa server berhasil memperoleh alamat pada antarmuka jaringan Host-Only Adapter, sehingga dapat diakses oleh komputer host, sebagaimana ditunjukkan pada Gambar 2. Pengujian konektivitas menggunakan perintah ping dari host ke alamat IP server menunjukkan hasil balasan (reply) yang konsisten tanpa paket yang hilang (0% loss) dengan waktu round-trip rata-rata di bawah 1 milidetik (Gambar 3), yang mengindikasikan bahwa jalur komunikasi antara client dan server telah siap digunakan untuk pengujian protokol Telnet maupun SSH.

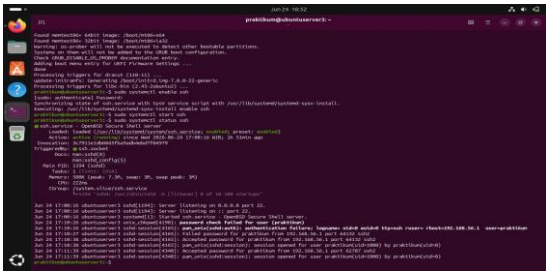


Gambar 2. Hasil pengecekan alamat IP Ubuntu Server menggunakan perintah ip a.

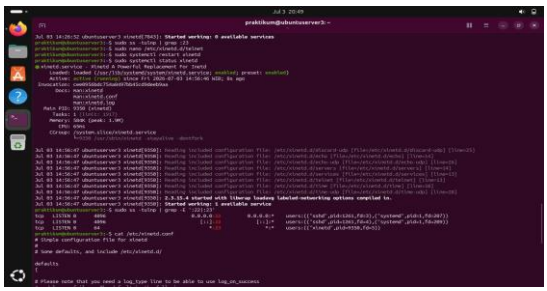


Gambar 3. Hasil pengujian konektivitas menggunakan perintah ping dari host ke server.

Layanan SSH yang diaktifkan melalui perintah systemctl enable ssh dan systemctl start ssh menunjukkan status active (running) dan siap menerima koneksi pada port 22, sebagaimana ditunjukkan oleh baris log "Server listening on 0.0.0.0 port 22" pada Gambar 4. Layanan Telnet yang dijalankan melalui super-server xinetd juga menunjukkan status active (running) setelah dilakukan konfigurasi pada berkas /etc/xinetd.d/telnet dan restart layanan xinetd, sebagaimana ditunjukkan pada Gambar 5. Verifikasi lebih lanjut menggunakan perintah ss -tulnp | grep -E ':22|:23' mengonfirmasi bahwa kedua port, yaitu port 22 untuk SSH dan port 23 untuk Telnet, telah berada pada kondisi listening dan siap menerima koneksi masuk. Dengan demikian, kedua layanan telah berjalan dengan baik dan siap diuji melalui koneksi client.



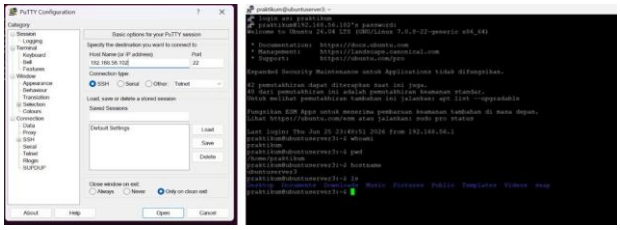
Gambar 4. Status layanan SSH yang aktif (active/running) pada port 22.



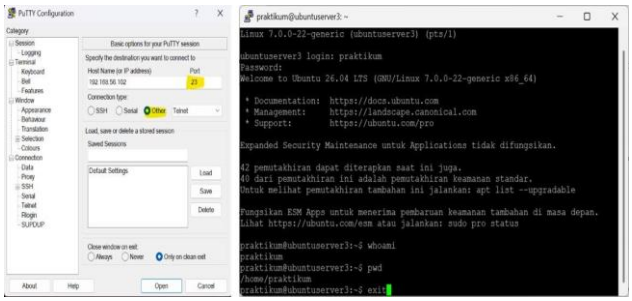
Gambar 5. Status layanan xinetd/Telnet yang aktif (active/running) pada port 23.

B. Hasil Pengujian Koneksi

Pengujian koneksi menggunakan aplikasi PuTTY menunjukkan bahwa proses autentikasi pada kedua protokol dapat dilakukan menggunakan akun praktikum yang telah dibuat sebelumnya. Pada sesi SSH (port 22), pengguna memasukkan alamat IP server dan memilih connection type SSH pada PuTTY Configuration, kemudian berhasil login dan menjalankan perintah dasar seperti whoami, pwd, hostname, dan ls tanpa kendala, sebagaimana ditunjukkan pada Gambar 6. Pada sesi Telnet (port 23), pengguna memilih connection type Other dengan protokol Telnet pada PuTTY Configuration, dan proses login serta eksekusi perintah yang sama juga dapat dilakukan dengan lancar, sebagaimana ditunjukkan pada Gambar 7. Secara fungsional, kedua protokol mampu memberikan akses remote yang setara dan tidak menunjukkan kendala teknis pada sisi antarmuka pengguna; perbedaan signifikan baru terlihat pada tahap analisis lalu lintas jaringan menggunakan Wireshark.



Gambar 6. Konfigurasi koneksi dan proses login SSH melalui aplikasi PuTTY pada port 22.



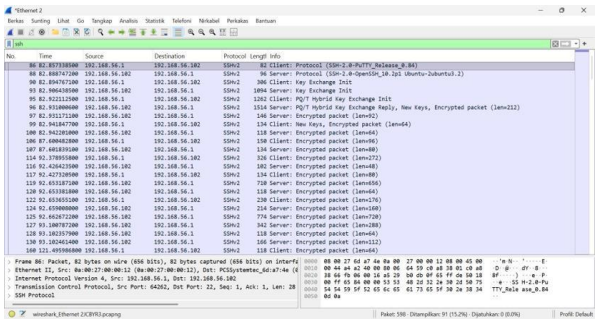
Gambar 7. Konfigurasi koneksi dan proses login Telnet melalui aplikasi PuTTY pada port 23.

C. Analisis Lalu Lintas Jaringan dengan Wireshark

Wireshark digunakan sebagai network protocol analyzer yang mampu menangkap seluruh paket data yang melintasi antarmuka jaringan, termasuk payload di dalamnya, sehingga sering dimanfaatkan baik untuk troubleshooting maupun pengujian keamanan melalui teknik sniffing [13]. Kemampuan Wireshark dalam merekonstruksi sesi

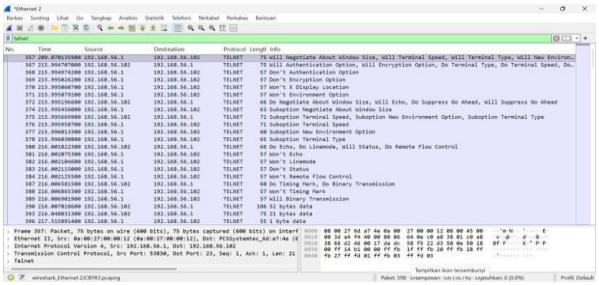
komunikasi, misalnya melalui fitur Follow TCP Stream, memungkinkan seorang penyadap membaca kembali seluruh isi komunikasi apabila protokol yang digunakan tidak menerapkan enkripsi, sebagaimana telah dibuktikan pada berbagai pengujian keamanan jaringan berbasis HTTP maupun layanan tanpa enkripsi lainnya [14].

Analisis menggunakan filter ssh dan tcp.port == 22 pada sesi SSH menunjukkan rangkaian paket berupa negosiasi protokol (Protocol SSH-2.0), key exchange initialization, proses hybrid key exchange, hingga paket-paket dengan label Encrypted packet, sebagaimana ditunjukkan pada Gambar 8. Tidak ditemukan satu pun paket yang menampilkan kredensial login maupun perintah dalam bentuk teks yang dapat dibaca, karena seluruh muatan data telah dienkripsi sejak tahap awal negosiasi kunci, sesuai dengan prinsip kerahasiaan (confidentiality) yang menjadi tujuan utama arsitektur SSH [6].



Gambar 8. Hasil capture Wireshark pada sesi SSH dengan filter ssh, menampilkan proses key exchange dan paket terenkripsi.

Sebaliknya, analisis menggunakan filter telnet dan tcp.port == 23 pada sesi Telnet menunjukkan paket-paket berupa negosiasi opsi terminal (Will/Won't/Do/Don't Negotiate) yang diikuti oleh paket data, sebagaimana ditunjukkan pada Gambar 9. Berbeda dengan SSH, muatan data pada paket Telnet tidak dienkripsi sehingga karakter yang diketikkan pengguna, termasuk kredensial autentikasi dan perintah yang dijalankan pada terminal, berpotensi dapat direkonstruksi oleh pihak yang melakukan penyadapan pada segmen jaringan yang sama. Ringkasan indikator hasil pengamatan kedua protokol disajikan pada Tabel I.



Gambar 9. Hasil capture Wireshark pada sesi Telnet dengan filter telnet, menampilkan paket negosiasi opsi terminal tanpa enkripsi.

TABEL I. Hasil Pengamatan Trafik Telnet dan SSH

Indikator Pengamatan	Telnet (port 23)	SSH (port 22)
Jenis paket dominan	TELNET (plaintext)	SSHv2 (Key Exchange, Encrypted)
Keterbacaan payload	Dapat dibaca langsung	Tidak dapat dibaca (ciphertext)
Negosiasi keamanan	Tidak ada	Ada (algorithm & key exchange)
Potensi kebocoran kredensial	Tinggi	Sangat rendah

D. Perbandingan Karakteristik Telnet dan SSH

Berdasarkan hasil implementasi dan analisis lalu lintas jaringan, perbandingan menyeluruh karakteristik kedua protokol dirangkum pada Tabel II.

TABEL II. Perbandingan Karakteristik Telnet dan SSH

Aspek	Telnet	SSH
Port default	23	22
Enkripsi data	Tidak tersedia	Tersedia
Keamanan login	Rendah	Tinggi
Risiko penyadapan	Tinggi	Rendah
Rekomendasi modern	Tidak disarankan	Sangat disarankan

Tabel II menegaskan bahwa perbedaan mendasar antara Telnet dan SSH terletak pada penerapan mekanisme enkripsi. Ketiadaan enkripsi pada Telnet menyebabkan seluruh proses komunikasi, termasuk autentikasi, rentan terhadap serangan pasif seperti sniffing maupun serangan aktif seperti session

hijacking. Kriptosistem simetris yang diterapkan pada sesi SSH modern juga memungkinkan pertukaran kunci sesi dilakukan secara aman tanpa perlu mendistribusikan kunci rahasia melalui saluran terbuka [9]. Pengujian kelayakan implementasi SSH sebagai pengganti protokol tanpa enkripsi pada layanan transfer file turut menegaskan bahwa penerapan SSH secara konsisten menurunkan risiko penyadapan kredensial dibandingkan protokol plaintext [10].

Meskipun demikian, keamanan SSH secara default tetap perlu diperkuat karena masih berpotensi menjadi sasaran serangan brute-force terhadap kredensial login [2]. Beberapa penelitian menunjukkan bahwa penerapan mekanisme tambahan seperti port knocking dapat meningkatkan keamanan port SSH dengan menyembunyikan port dari proses port scanning sebelum client melakukan autentikasi [11]. Pendekatan mitigasi lain seperti pembatasan percobaan login dan intrusion prevention pada perangkat jaringan juga terbukti efektif menurunkan tingkat keberhasilan serangan brute-force terhadap layanan remote access [12]. Temuan serupa pada pengujian keamanan jaringan nirkabel turut menegaskan bahwa protokol maupun layanan tanpa mekanisme autentikasi dan enkripsi yang memadai secara konsisten lebih mudah disusupi melalui teknik packet sniffing dibandingkan layanan yang telah menerapkan proteksi kriptografis [15]. Dengan mempertimbangkan seluruh temuan tersebut, penggunaan Telnet pada lingkungan produksi saat ini sudah tidak relevan dan sebaiknya dibatasi hanya pada kebutuhan pembelajaran maupun simulasi laboratorium yang terisolasi dari jaringan produksi, sedangkan SSH tetap menjadi pilihan utama dan standar de facto untuk kebutuhan administrasi server jarak jauh pada lingkungan jaringan modern.

E. Kendala dan Solusi Penelitian

Selama pelaksanaan penelitian, ditemukan beberapa kendala teknis. Pertama, terdapat perbedaan perilaku layanan Telnet pada Ubuntu Server versi terbaru dibandingkan referensi yang digunakan, sehingga proses konfigurasi memerlukan penyesuaian pada berkas konfigurasi xinetd. Kedua, beberapa hasil pengujian, seperti verifikasi port Telnet, berbeda dengan contoh pada modul referensi sehingga memerlukan pengecekan ulang terhadap konfigurasi xinetd dan layanan Telnet. Solusi yang diterapkan meliputi pengecekan status layanan secara berkala menggunakan systemctl serta memastikan konfigurasi Telnet pada direktori /etc/xinetd.d/ telah sesuai, kemudian melakukan restart layanan xinetd setelah konfigurasi diubah dan memverifikasi kembali bahwa port 23 telah berada pada kondisi listening menggunakan perintah ss -tulnp. Pengalaman ini menegaskan pentingnya menyesuaikan referensi konfigurasi dengan versi sistem operasi yang digunakan pada saat penelitian dilaksanakan.

IV. KESIMPULAN

Berdasarkan hasil implementasi dan analisis yang telah dilakukan, dapat disimpulkan bahwa Telnet dan SSH sama-sama mampu menyediakan layanan akses jarak jauh ke server, namun keduanya memiliki perbedaan yang signifikan dari aspek keamanan. Hasil pengamatan menggunakan Wireshark menunjukkan bahwa SSH menerapkan mekanisme enkripsi sejak tahap negosiasi kunci hingga seluruh sesi komunikasi berlangsung, sehingga kredensial dan data yang dipertukarkan tidak dapat dibaca oleh pihak yang melakukan penyadapan. Sebaliknya, Telnet mengirimkan seluruh data dalam bentuk plain text sehingga sangat rentan terhadap penyadapan, sebagaimana dibuktikan melalui hasil capture yang menampilkan karakter demi karakter dari sesi komunikasi secara terbuka.

Dengan demikian, tujuan penelitian ini untuk membuktikan perbedaan tingkat keamanan Telnet dan SSH telah tercapai. SSH lebih direkomendasikan untuk digunakan dalam administrasi server modern karena mampu menjaga kerahasiaan dan integritas data selama proses komunikasi, sedangkan Telnet hanya layak digunakan sebagai media pembelajaran atau simulasi pada lingkungan

laboratorium yang terisolasi dari jaringan produksi. Bagi administrator jaringan, temuan ini menegaskan pentingnya migrasi penuh dari Telnet menuju SSH pada seluruh lini layanan akses jarak jauh, disertai penerapan praktik keamanan tambahan seperti autentikasi berbasis kunci publik, pembatasan percobaan login, serta pemantauan log secara berkala guna memperkuat pertahanan terhadap ancaman brute-force.

Penelitian lanjutan dapat dikembangkan dengan menambahkan skenario serangan aktif, seperti simulasi Man-in-the-Middle (MITM) atau brute-force menggunakan perangkat penetration testing, serta penerapan mekanisme pengamanan tambahan seperti port knocking, fail2ban, maupun autentikasi berbasis kunci publik, untuk mengukur secara lebih mendalam tingkat ketahanan (robustness) SSH dibandingkan Telnet pada kondisi jaringan yang lebih kompleks dan mendekati lingkungan produksi sesungguhnya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bapak Ridwan Dwi Irawan, M.Kom. selaku dosen pengampu mata kuliah Keamanan Jaringan, serta kepada Fakultas Ilmu Komputer Universitas Duta Bangsa Surakarta atas fasilitas dan bimbingan yang diberikan selama pelaksanaan praktikum dan penyusunan artikel ini.

REFERENSI

[1] N. Fadillah, dkk., "Analisis Keamanan Protokol Telnet vs SSH Menggunakan Wireshark Dalam Jaringan Terbuka," Journal of System & Technology (SYSTEC), 2025.

[2] J. Park, J. Kim, B. B. Gupta, and N. Park, "Network Log-Based SSH Brute-Force Attack Detection Model," Computers, Materials and Continua, 2023.

[3] A. S. Syahab, E. I. H. Ujianto, and Rianto, "Penggunaan Wireshark dan NESSUS untuk Analisis SSL/TLS pada Keamanan Data Pengguna Website," Jurnal Informatika, vol. 7, no. 2, pp. 183-192, 2023.

[4] W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed. Boston, USA: Pearson, 2017.

[5] J. Postel and J. Reynolds, "Telnet Protocol Specification," RFC 854, Internet Engineering Task Force, 1983.

[6] T. Ylonen and C. Lonvick, Eds., "The Secure Shell (SSH) Protocol Architecture," RFC 4251, Internet Engineering Task Force, 2006.

[7] D. J. Barrett, R. E. Silverman, and R. G. Byrnes, SSH, The Secure Shell: The Definitive Guide, 2nd ed. Sebastopol, USA: O'Reilly, 2005.

[8] A. R. Mubaraq, N. A. Verdikha, and M. T. Sumadi, "Analisis Kualitas Jaringan Internet Wireless LAN PT. Teladan Prima Agro," Jurnal Publikasi Teknik Informatika, vol. 3, no. 1, pp. 38-47, 2023.

[9] M. Iqbal, "Keamanan Remote Server Melalui SSH Dengan Kriptosistem Simetris," TECHSI: Jurnal Penelitian Teknik Informatika, vol. 3, no. 2, pp. 54-66, 2013.

[10] B. Sakti, A. Aziz, and A. Doewes, "Uji Kelayakan Implementasi SSH sebagai Pengaman FTP Server dengan Penetration Testing," Jurnal ITSMART, vol. 2, no. 1, pp. 44-51, 2013.

[11] D. Desmira and R. Wiryadinata, "Rancang Bangun Keamanan Port Secure Shell (SSH) Menggunakan Metode Port Knocking," Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI), vol. 5, no. 1, pp. 28-33, 2022.

[12] B. Arifwidodo, Y. Syuhada, and S. Ikhwan, "Analisis Kinerja Mikrotik Terhadap Serangan Brute Force Dan DDoS," Techno.Com, vol. 20, no. 3, pp. 392-399, 2021.

[13] R. M. Farhan and G. H. A. Kusuma, "Teknik Sniffing Jaringan Menggunakan Wireshark," Journal of Informatics and Advanced Computing (JIAC), vol. 4, no. 1, pp. 87-93, 2023.

[14] Z. M. Luthfansa and U. D. Rosiani, "Pemanfaatan Wireshark untuk Sniffing Komunikasi Data Berprotokol HTTP pada Jaringan Internet," Journal of Information Engineering and Educational Technology, vol. 5, no. 1, pp. 34-39, 2021.

[15] Fatimah, T. Mary, and A. Y. Pernanda, "Analisis Keamanan Jaringan Wi-Fi Terhadap Serangan Packet Sniffing di Universitas PGRI Sumatera Barat," JURTEII: Jurnal Teknologi Informasi, vol. 1, no. 2, pp. 7-11, 2022.