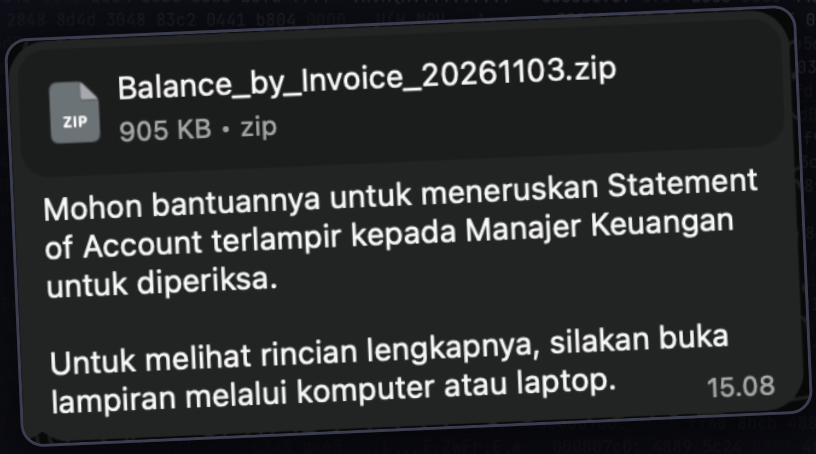
liupurnomo

EBOOK GRATIS

YANG ANDA TERIMA BUKAN FAKTUR

Waspada Lampiran ZIP di WhatsApp, Apa Isinya, dan ke Mana Data Anda Dikirim



Liu Purnomo, S.Kom.

liupurnomo.com

liupurnomo

Yang Anda Terima Bukan Faktur

Waspada Lampiran ZIP di WhatsApp, Apa Isinya, dan ke Mana Data Anda Dikirim

Penulis: Liu Purnomo, S.Kom.
Edisi pertama, November 2026

Ebook ini dibagikan gratis sebagai upaya edukasi publik agar tidak ada lagi korban penipuan lewat lampiran palsu.

Boleh disalin dan dibagikan seluas-luasnya dalam bentuk utuh tanpa diubah dan tanpa dipungut biaya. Tidak untuk diperjualbelikan.

Situs: liupurnomo.com
Email: hi@liupurnomo.com

Seluruh analisis dilakukan secara statis tanpa menjalankan program jahat. Foto profil, nama, dan nomor pengirim pada contoh pesan telah disamarkan. Alamat server dan nilai *hash* dicantumkan untuk keperluan deteksi dan pelaporan.



Kata Pengantar

Penipuan lewat WhatsApp terus berganti rupa. Beberapa tahun lalu yang banyak beredar adalah undangan pernikahan, surat tilang, dan resi paket dalam bentuk file APK yang menyasar ponsel Android. Kini muncul gelombang yang menyasar komputer kantor. Bentuknya pesan sopan berisi lampiran ZIP yang mengaku sebagai faktur, saldo tagihan, atau *statement of account*, dengan permintaan agar file diteruskan kepada manajer keuangan dan dibuka di komputer.

Nama file, nomor pengirim, dan isi lampirannya terus berganti. Satu hal tidak pernah berubah, yaitu cara kerjanya. Karena itu buku ini tidak berfokus pada satu file tertentu. Buku ini menjelaskan **model kejahatannya**, yaitu enam tahap yang selalu dilalui serangan semacam ini, dari umpan sampai panen. Dengan memahami model tersebut, pembaca dapat mengenali varian baru yang belum pernah dilihat sebelumnya.

Sebagai bukti, buku ini membedah dua sampel nyata yang diterima pada Oktober dan November 2026. Keduanya dibongkar sampai lapisan terdalam tanpa pernah dijalankan. Hasilnya memperlihatkan dua cara penyamaran yang berbeda, tetapi mengikuti pola yang sama persis, lengkap dengan alamat server tujuan pengiriman data korban.

Isi buku dibagi menjadi empat bagian. Bagian I menjelaskan model kejahatannya dan ditujukan bagi semua pembaca. Bagian II menguraikan hasil pembedahan kedua sampel. Bagian III menunjukkan langkah investigasi yang bisa diulang sendiri secara aman. Bagian IV berisi langkah pencegahan dan penanganan bagi individu maupun perusahaan.

Pembaca yang tidak berlatar belakang teknis cukup membaca Bagian I dan Bagian IV. Pembaca dari bidang teknologi informasi dapat memakai Bagian II, Bagian III, dan lampiran indikator teknis sebagai bahan deteksi dan pelatihan.

Jakarta, November 2026

Liu Purnomo, S.Kom.

Daftar Isi

Kata Pengantar 3

Daftar Gambar 8

Daftar Tabel 9

Bagian I — Mengenali Modusnya 10

Bab 1 — Satu Pola, Banyak Wajah 11

1.1 Lima Tanda Bahaya 13

1.2 Mengapa Sasarannya Orang Kantor 15

1.3 Psikologi di Balik Naskah 15

1.4 Bukan Kejadian Tunggal 16

Bab 2 — Model Kejahatannya 17

2.1 Tahap 1, Umpan 18

2.2 Tahap 2, Pembawa 19

2.3 Tahap 3, Pembuka 19

2.4 Tahap 4, Muatan 20

2.5 Tahap 5, Kendali 20

2.6 Tahap 6, Panen 21

2.7 Yang Berubah dan yang Tetap 21

2.8 Ringkasan Temuan 22

Bagian II — Membedah Dua Sampel 23

Bab 3 — Sampel November, Tujuh Lapis Penyamaran 24

3.1 Lapis 1, Pesan dan ZIP 26

3.2 Lapis 2, Program Asli Bertanda Tangan 26

3.3 Lapis 3, DLL Palsu yang Menyamar 27

3.4 Lapis 4, Stub Shellcode 28

3.5 Lapis 5, RC4 yang Dimodifikasi 29

3.6 Lapis 6, Program yang Kepalanya Disamarkan 31

3.7 Lapis 7, Konfigurasi Terenkripsi 32

Bab 4 — Ke Mana Perginya Sampel November 33

4.1 Menemukan Konfigurasi 33
4.2 Kunci yang Terlalu Sederhana 33
4.3 Isi Konfigurasi 34
4.4 Siapa Pemilik Alamat Itu 35
4.5 Apa yang Terjadi Setelah Terhubung 36

Bab 5 — Sampel Oktober, Muatan di Dalam Tanda Tangan 37

5.1 Dua File dengan Tanda Tangan Resmi 38
5.2 DLL Asli yang Ditambal 39
5.3 Membaca Dirinya Sendiri 40
5.4 Muatan di Area Tanda Tangan Digital 40
5.5 Tahap Dua, Pengunduh 41
5.6 Ke Mana Perginya 41
5.7 Dua Sampel, Satu Model 41

Bab 6 — Kemampuan Program Inti 43

6.1 Membaca Daftar Impor 44
6.2 Teks yang Disembunyikan 44
6.3 Perilaku yang Bisa Disimpulkan 45

Bab 7 — Apa yang Akan Diambil 46

7.1 Data yang Diincar 47
7.2 Ujung yang Paling Merugikan 48
7.3 Korban Berikutnya 49
7.4 Pemetaan ke MITRE ATT&CK 49

Bagian III — Menyelidiki Sendiri dengan Aman 51

Bab 8 — Prinsip dan Persiapan 52

8.1 Analisis Statis dan Dinamis 52
8.2 Aturan Keselamatan 53
8.3 Alat yang Dibutuhkan 54

Bab 9 — Langkah Investigasi 55

9.1 Langkah 1, Mengenali Jenis File 55
9.2 Langkah 2, Mencatat Sidik Digital 55
9.3 Langkah 3, Membaca Identitas Program 56
9.4 Langkah 4, Mencari Teks yang Terbaca 56
9.5 Langkah 5, Mengukur Entropi 57

9.6 Langkah 6, Memetakan Struktur File .bin 58
9.7 Langkah 7, Membongkar Kode Stub 58
9.8 Langkah 8, Mendekripsi Muatan 59
9.9 Langkah 9, Membaca Kemampuan Program Inti 61
9.10 Langkah 10, Menemukan Alamat Server 62
9.11 Langkah 11, Memeriksa Area Tanda Tangan 63
9.12 Langkah 12, Membuka Tahap Dua 63

Bab 10 — Memakai dan Melaporkan Temuan 65
10.1 Indikator Kompromi 65
10.2 Memakai IOC di Kantor 66
10.3 Contoh Aturan YARA 66
10.4 Melaporkan Server Pelaku 67
10.5 Kapan Perlu Analisis Dinamis 67

Bagian IV — Mencegah dan Menangani 69

Bab 11 — Langkah Pencegahan 70
11.1 Kebiasaan Pribadi 70
11.2 Pengaturan WhatsApp 71
11.3 Pengaturan Windows 71
11.4 Prosedur Kantor dan Bagian Keuangan 72
11.5 Perlindungan Teknis untuk Tim TI 73
11.6 Pohon Keputusan Sebelum Membuka Lampiran 74

Bab 12 — Bila Terlanjur Membuka 75
12.1 Menit Pertama 76
12.2 Jam Pertama 76
12.3 Hari Pertama 76
12.4 Pemeriksaan oleh Tim TI 77
12.5 Sesudahnya 77
12.6 Ke Mana Melapor 78

Lampiran 79

Lampiran A — Daftar Periksa Lampiran Chat 80

Lampiran B — Indikator Teknis Lengkap 82

Glosarium 85

Daftar Pustaka 88

Tentang Penulis 89

Daftar Gambar

Gambar 1.1 Dua pesan yang diterima dengan selang 26 hari 12
Gambar 1.2 Lima tanda bahaya pada pesan November 14
Gambar 2.1 Model kejahatan enam tahap 18
Gambar 3.1 Isi lampiran sampel November dan peran masing-masing file 24
Gambar 3.2 Tujuh lapis penyamaran dalam sampel 25
Gambar 3.3 Susunan isi file SbieDll.bin hasil pembedahan 28
Gambar 3.4 Entropi SbieDll.bin per kilobyte sebelum dan sesudah didekripsi 30
Gambar 4.1 Arah komunikasi antara komputer korban dan server penipu 35
Gambar 5.1 Rantai pemuatan sampel Oktober 38
Gambar 6.1 Sebaran fungsi Windows yang diimpor program inti menurut kelompok kemampuan 43
Gambar 7.1 Data dan akses yang umumnya diincar program jahat jenis ini 47
Gambar 8.1 Prinsip lingkungan analisis yang aman 53
Gambar 11.1 Pohon keputusan sebelum membuka lampiran 74
Gambar 12.1 Urutan langkah darurat setelah file terlanjur dibuka 75

Daftar Tabel

Tabel 1.1 Perbandingan dua pesan yang diterima 13
Tabel 1.2 Lima tanda bahaya yang selalu muncul 14
Tabel 1.3 Teknik rekayasa sosial dalam naskah pesan 15
Tabel 2.1 Cara menyembunyikan muatan 20
Tabel 2.2 Unsur yang berubah dan yang tetap 21
Tabel 2.3 Ringkasan hasil pembedahan dua sampel 22
Tabel 3.1 Identitas asli file umpan 26
Tabel 3.2 Fungsi Windows yang terlihat pada DLL palsu 27
Tabel 3.3 Perbandingan RC4 standar dan versi sampel 30
Tabel 3.4 Identitas program inti 31
Tabel 4.1 Isi konfigurasi setelah didekripsi 34
Tabel 5.1 Isi lampiran sampel Oktober 38
Tabel 5.2 Perbandingan teknis dua sampel 41
Tabel 6.1 Kelompok fungsi impor dan artinya 44
Tabel 6.2 Teks yang dibuka dari fungsi utama 45
Tabel 7.1 Data yang diincar dan akibatnya 47
Tabel 7.2 Pemetaan teknik ke MITRE ATT&CK 49
Tabel 8.1 Perbandingan analisis statis dan dinamis 52
Tabel 8.2 Alat analisis yang dipakai 54
Tabel 10.1 Indikator kompromi utama 65
Tabel 11.1 Pengaturan WhatsApp yang disarankan 71
Tabel 11.2 Prosedur yang disarankan untuk kantor 72
Tabel 11.3 Kontrol teknis terhadap tiap lapis serangan 73
Tabel 12.1 Jejak yang perlu diperiksa 77
Tabel 12.2 Saluran pelaporan 78
Tabel A.1 Daftar periksa sebelum membuka lampiran 80
Tabel A.2 Bila terlanjur membuka 81
Tabel B.1 Sidik digital sampel November (SHA-256) 82
Tabel B.2 Sidik digital sampel Oktober (SHA-256) 82
Tabel B.3 Indikator jaringan dan sistem 83
Tabel B.4 Parameter teknis untuk penyelidik 83

Bagian I

MENGENALI MODUSNYA

Satu pola dengan banyak wajah. Model kejahatan enam tahap yang perlu dikenali semua orang.

> **“Only amateurs attack machines; professionals target people.”**
>
> *Hanya amatir yang menyerang mesin. Para profesional menyasar manusia.*
>
> Bruce Schneier, Crypto-Gram, 15 Oktober 2000



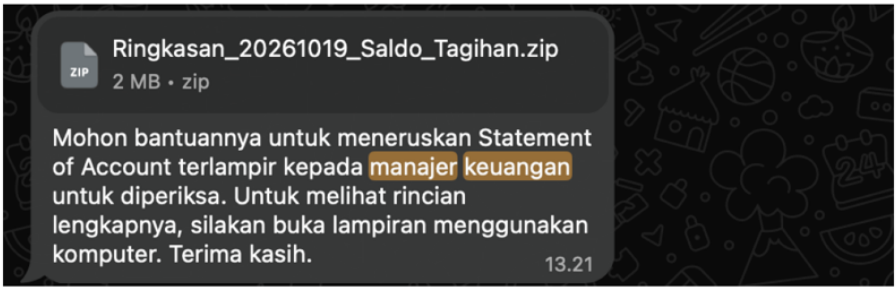
Bab 1

Satu Pola, Banyak Wajah

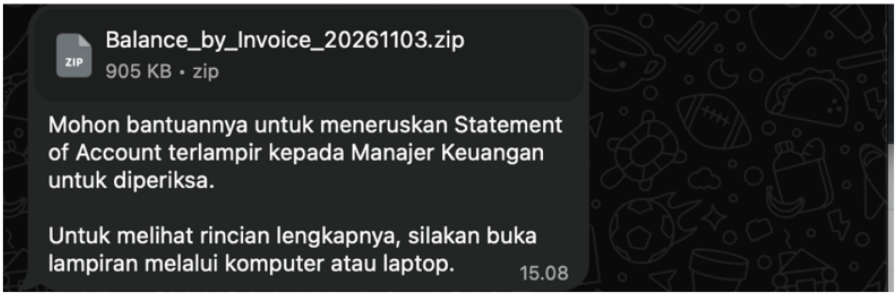
Pada 21 Oktober 2026 pukul 13.21, sebuah pesan masuk dari nomor berkode negara +91 (India). Isinya satu lampiran bernama **Ringkasan_20261019_Saldo_Tagihan.zip** berukuran 2 MB, disertai kalimat sopan yang meminta lampiran diteruskan kepada manajer keuangan.

Dua puluh enam hari kemudian, pada 16 November 2026 pukul 15.08, pesan serupa datang lagi. Kali ini dari nomor Indonesia berawalan +62, dengan lampiran **Balance_by_Invoice_20261103.zip** berukuran 905 KB.

21 Oktober 2026, nomor +91 (India), Ringkasan_20261019_Saldo_Tagihan.zip, 2 MB

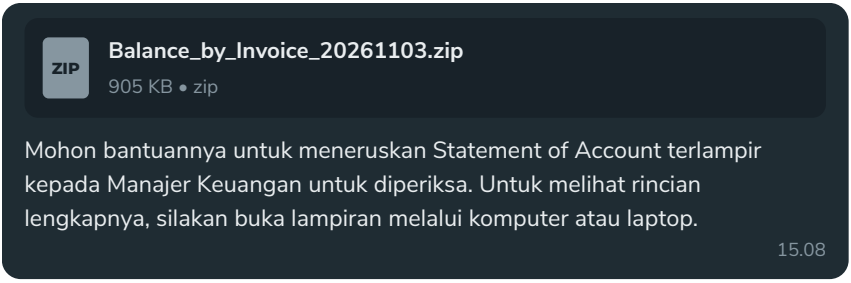


16 November 2026, nomor +62 (Indonesia), Balance_by_Invoice_20261103.zip, 905 KB



Gambar 1.1 Dua pesan yang diterima dengan selang 26 hari. Foto profil dan nomor pengirim disamarkan.

Bila dibaca berdampingan, kalimat kedua pesan nyaris identik kata per kata.



Nomor pengirim berbeda negara, nama file berbeda, ukuran berbeda, dan seperti yang akan dibuktikan di Bagian II, isi lampirannya pun dibuat dengan

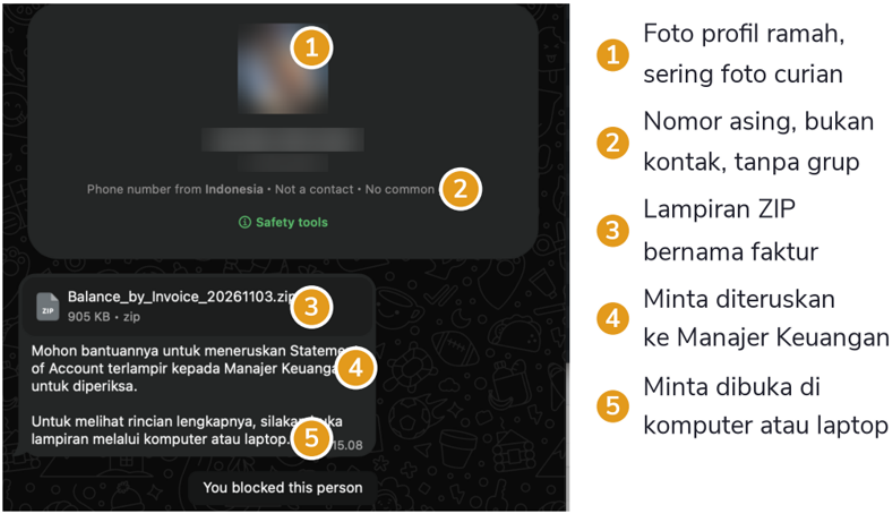
teknik yang berbeda. Namun naskah pesannya sama. Ini tanda bahwa kedua pesan berasal dari satu model kejahatan yang sama, dijalankan berulang-ulang dengan bahan yang terus diganti.

Tabel 1.1 Perbandingan dua pesan yang diterima

Unsur	Pesan Oktober	Pesan November
Tanggal	21 Oktober 2026, 13.21	16 November 2026, 15.08
Nomor pengirim	+91, India	+62, Indonesia
Nama lampiran	Ringkasan_20261019_ Saldo_Tagihan.zip	Balance_by_Invoice_ 20261103.zip
Ukuran	2 MB	905 KB
Naskah pesan	Statement of Account, manajer keuangan, buka di komputer	Sama
Isi ZIP	Dua file, program NetEase dan DLL yang ditambal	Tiga file, program Sandboxie, DLL palsu, dan muatan .bin
Tujuan akhir	Mengunduh tahap berikutnya dari blasidha.eu.cc	Terhubung ke server 43.156.35.124

1.1 Lima Tanda Bahaya

Walaupun bahannya berganti, pesan-pesan seperti ini selalu membawa tanda bahaya yang sama.



Gambar 1.2 Lima tanda bahaya pada pesan November. Pesan Oktober memuat tanda yang sama.

Tabel 1.2 Lima tanda bahaya yang selalu muncul

Tanda	Mengapa mencurigakan
Pengirim tidak dikenal	WhatsApp sendiri menampilkan keterangan *Not a contact* dan *No common groups*. Nomornya bisa dari dalam maupun luar negeri dan selalu berganti.
Foto profil meyakinkan	Foto orang yang tampak ramah atau berpenampilan kantoran menurunkan kewaspadaan. Foto seperti ini sering diambil dari media sosial orang lain tanpa izin.
Lampiran berupa arsip	Dokumen keuangan yang wajar dikirim dalam bentuk PDF. ZIP, RAR, IMG, atau VHD dipakai untuk membungkus program agar tidak langsung terlihat.
Minta diteruskan ke atasan	Penerima dijadikan kurir agar file sampai ke orang yang memegang akses keuangan.
Minta dibuka di komputer	Isi lampiran hanya berjalan di Windows. Di ponsel file itu tidak berbuat apa-apa, sehingga pelaku membutuhkan korban yang membukanya di komputer.

1.2 Mengapa Sasarannya Orang Kantor

Model ini mengincar komputer kantor, terutama milik bagian keuangan, administrasi, dan pembelian. Komputer seperti ini biasanya menyimpan akses ke *internet banking* perusahaan, email resmi, data pemasok, serta daftar tagihan yang belum dibayar. Satu komputer keuangan yang berhasil dikuasai bisa bernilai jauh lebih besar daripada ratusan ponsel pribadi.

Nomor WhatsApp staf kantor mudah ditemukan. Banyak yang tercantum di situs perusahaan, kartu nama, katalog produk, profil LinkedIn, atau grup pemasok. Pelaku cukup mengirim naskah yang sama ke ratusan nomor dan menunggu satu orang mengklik.

Ingat

Pelaku tidak perlu menipu Manajer Keuangan secara langsung. Ia cukup meyakinkan satu staf untuk meneruskan file. Pesan yang datang dari rekan sendiri jauh lebih dipercaya daripada pesan dari nomor asing.

1.3 Psikologi di Balik Naskah

Naskah pesan disusun dengan memanfaatkan kecenderungan manusia yang sudah lama dikenal dalam kajian rekayasa sosial (*social engineering*).

Tabel 1.3 Teknik rekayasa sosial dalam naskah pesan

Teknik	Wujudnya dalam pesan
Kewajaran	Memakai istilah keuangan yang lazim sehingga pesan terasa seperti urusan kantor biasa.
Otoritas	Menyebut jabatan Manajer Keuangan agar penerima merasa urusan ini penting dan harus segera diteruskan.
Permintaan kecil	Hanya meminta file diteruskan. Permintaan kecil lebih mudah dituruti daripada permintaan besar.

Teknik	Wujudnya dalam pesan
Kesopanan	Kalimat diawali “Mohon bantuannya” dan ditutup “Terima kasih”. Orang Indonesia cenderung sungkan menolak permintaan yang disampaikan dengan sopan.
Instruksi teknis	Saran membuka di komputer terdengar membantu, padahal mengarahkan korban ke perangkat yang bisa diserang.

1.4 Bukan Kejadian Tunggal

Dua pesan di atas adalah bagian dari gelombang yang lebih besar. Pada September 2026, peneliti Seqrite Labs dari Quick Heal Technologies melaporkan kampanye program jahat lewat WhatsApp yang menyasar tim keuangan, eksekutif senior, dan akuntan. Lampirannya menyamar sebagai dokumen berjudul *Financial Report*, *Account Statement*, *Outstanding Payment List*, dan *Debt Confirmation*.

Laporan yang sama mencatat bahwa bentuk lampirannya terus berkembang. Awalnya berupa skrip .vbs, lalu berganti menjadi arsip ZIP berisi program dan file pendamping yang memakai *DLL side-loading*, persis seperti dua sampel di buku ini. Ada pula varian berupa file .img dan .vhd yang oleh Windows diperlakukan seperti *disk* baru. Komputer yang terinfeksi bahkan dapat memakai sesi WhatsApp Web korban untuk meneruskan file yang sama secara otomatis kepada kontaknya.

⚠ Pelajaran utama

Yang perlu dikenali adalah polanya. Nama file, nomor, dan teknik penyamaran akan terus berganti, sedangkan pola enam tahap yang dijelaskan pada bab berikutnya tetap sama.



Bab 2

Model Kejahatannya

Serangan seperti ini dijalankan layaknya sebuah usaha. Ada bagian yang menyiapkan naskah dan nomor, ada bagian yang merakit lampiran, ada server yang menampung data curian, dan ada pihak yang mengubah data curian menjadi uang. Setiap bagian bisa diganti tanpa mengubah bagian lain. Itulah sebabnya bahan serangan terus berubah sementara alurnya tetap.



Isi kolom kanan selalu berganti. Urutan enam tahapnya tetap sama.

Gambar 2.1 Model kejahatan enam tahap. Contoh di kolom kanan berganti dari kampanye ke kampanye.

2.1 Tahap 1, Umpan

Umpan adalah pesan yang membuat korban mau bertindak. Temanya hampir selalu keuangan, karena dokumen keuangan wajar diterima orang kantor dari pihak luar. Bentuknya bisa faktur, saldo tagihan, *statement of account*, laporan keuangan, daftar pembayaran tertunggak, atau konfirmasi utang. Pada beberapa varian, pelaku bahkan menyamar sebagai lembaga resmi.

WhatsApp dipilih karena dianggap lebih personal daripada email dan tidak melewati penyaring lampiran milik kantor. Nomor pengirim mudah diganti dan bisa berasal dari negara mana pun.

2.2 Tahap 2, Pembawa

Pembawa adalah wadah yang membungkus program jahat agar lolos sampai ke komputer korban. Yang paling sering dipakai adalah arsip ZIP atau RAR. Varian lain memakai file *disk image* seperti .img dan .vhd, atau skrip .vbs.

- Arsip membuat file program tidak langsung terlihat sebagai program.
- Arsip menjaga beberapa file tetap bersama dalam satu folder, syarat penting bagi tahap berikutnya.
- Nama arsip bisa dibuat meyakinkan dengan deretan angka tanggal, sehingga terlihat seperti dokumen dari sistem akuntansi.
- Windows sering menyembunyikan ekstensi file, sehingga program bernama *Ringkasan_Faktur_Pemasok.exe* hanya tampil sebagai *Ringkasan_Faktur_Pemasok*.

2.3 Tahap 3, Pembuka

Pada tahap ini pelaku menunggangi program sah yang memiliki tanda tangan digital resmi. Pada sampel November, yang ditunggangi adalah komponen Sandboxie. Pada sampel Oktober, yang ditunggangi adalah program buatan NetEase beserta pustaka pelaporan galatnya. Kedua pengembang tersebut tidak terlibat. Program mereka dipakai tanpa izin karena tanda tangan digitalnya membuat Windows dan antivirus cenderung percaya.

Program sah itu punya kebiasaan memuat file pendamping berupa DLL dari folder tempat ia berada. Pelaku menaruh DLL palsu atau DLL yang sudah ditambal di folder yang sama. Ketika program sah dijalankan, ia memuat DLL tersebut tanpa curiga, dan kode jahat ikut berjalan seolah-olah bagian dari program yang sah. Teknik ini disebut *DLL side-loading*.

Perumpamaan

Bayangkan seorang kurir resmi berseragam lengkap yang diminta membawa sebuah kotak ke dalam gedung. Satpam membiarkannya masuk karena seragam dan kartu identitasnya asli. Isi kotak itu ternyata sudah ditukar orang lain di tengah jalan. Program bertanda tangan resmi adalah kurirnya. DLL palsu dan muatannya adalah isi kotak yang sudah ditukar.

2.4 Tahap 4, Muatan

Muatan adalah kode jahat yang sebenarnya. Pelaku berusaha keras menyembunyikannya agar tidak dikenali antivirus. Dua sampel di buku ini memakai tiga cara berbeda.

Tabel 2.1 Cara menyembunyikan muatan

Cara	Contoh pada sampel
File terenkripsi terpisah	Sampel November menyimpan muatan di file *SbieDll.bin* yang dienkripsi RC4 versi modifikasi.
Di dalam tanda tangan digital	Sampel Oktober menyelipkan muatan di area tanda tangan digital program NetEase. Area ini tidak ikut dihitung saat Windows memeriksa keaslian tanda tangan.
Diunduh belakangan	Muatan sampel Oktober hanyalah pengunduh kecil. Kode jahat utamanya diambil dari internet setelah komputer terinfeksi.

Kedua sampel juga menjalankan muatannya langsung di memori tanpa menyimpannya sebagai file biasa. Cara ini membuat pemindaian file di *harddisk* tidak menemukan apa pun.

2.5 Tahap 5, Kendali

Setelah muatan berjalan, komputer korban menghubungi server milik pelaku. Server ini biasanya berupa server awan sewaan atau domain gratis yang mudah dibuang dan diganti. Sampel November menghubungi alamat IP di

jaringan Tencent Cloud Singapura. Sampel Oktober mengunduh tahap berikutnya dari subdomain gratis berakhiran eu.cc. Lokasi server tidak menunjukkan lokasi atau kewarganegaraan pelaku, karena server seperti ini bisa disewa siapa pun dari mana pun.

2.6 Tahap 6, Panen

Tahap terakhir adalah mengubah akses menjadi keuntungan. Pelaku mencuri kata sandi dan sesi login, membaca email dan percakapan, mempelajari alur tagihan perusahaan, lalu membelokkan pembayaran ke rekening miliknya. Akun WhatsApp korban juga dipakai untuk mengirim umpan yang sama kepada kontak korban, sehingga siklus kembali ke tahap pertama. Rinciannya dibahas pada Bab 7.

2.7 Yang Berubah dan yang Tetap

Tabel 2.2 Unsur yang berubah dan yang tetap

Tahap	Yang berubah	Yang tetap
Umpan	Nomor, negara, foto profil, nama file	Tema keuangan, minta diteruskan ke atasan, minta dibuka di komputer
Pembawa	ZIP, RAR, IMG, VHD, VBS	Program dibungkus agar tidak terlihat sebagai program
Pembuka	Program sah yang ditunggangi	Program bertanda tangan resmi yang memuat DLL dari foldernya
Muatan	Lokasi dan cara enkripsi	Disembunyikan, lalu dijalankan di memori
Kendali	Alamat IP dan domain	Komputer korban menghubungi server pelaku
Panen	Data yang diambil	Uang perusahaan dan akses ke korban berikutnya

2.8 Ringkasan Temuan

Ringkasan berikut menjawab pertanyaan utama untuk kedua sampel. Pembaca yang tidak ingin masuk ke detail teknis dapat langsung melanjutkan ke Bagian IV.

Tabel 2.3 Ringkasan hasil pembedahan dua sampel

Pertanyaan	Sampel Oktober	Sampel November
Apa ini?	Pengunduh tersembunyi di dalam tanda tangan digital program NetEase, dijalankan oleh DLL yang ditambal	Program pengendali jarak jauh (*backdoor*) dalam tujuh lapis penyamaran
Ke mana arahnya?	Mengunduh tahap berikutnya dari **blasidha.eu.cc**	Terhubung ke **43.156.35.124** port **1234**, Tencent Cloud Singapura
Apa yang diambil?	Bergantung pada tahap yang diunduh, umumnya kendali penuh atas komputer	Kendali penuh atas komputer, termasuk data, sesi login, dan jaringan kantor
Waktu pembuatan	Program induk dikompilasi 11 Agustus 2026, paket dirakit 21 Oktober 2026	Program inti dikompilasi 6 Maret 2026, paket dirakit 16 November 2026

Bagian II

MEMBEDAH DUA SAMPEL

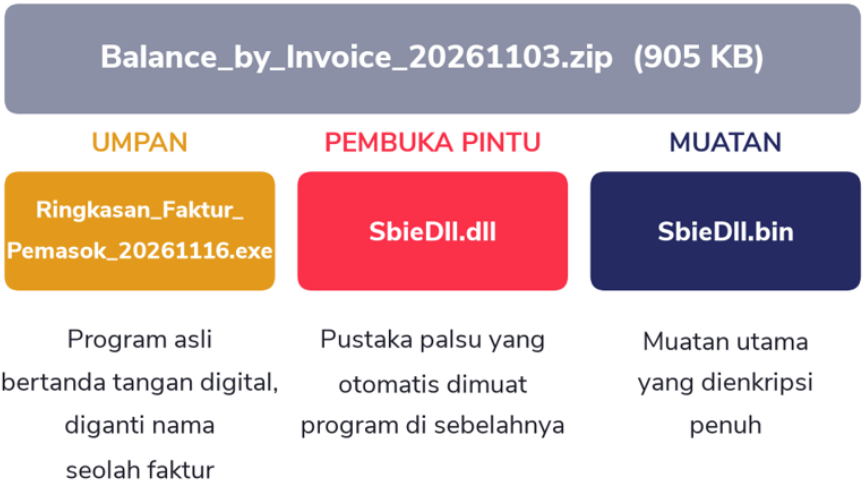
Dua sampel nyata dibongkar sampai lapisan terdalam, lengkap dengan alamat tujuannya.

> **“If it’s smart, it’s vulnerable.”**
>
> *Bila sesuatu itu pintar, ia pasti punya celah.*
>
> Mikko Hyppönen, judul bukunya, 2022

Bab 3

Sampel November, Tujuh Lapis Penyamaran

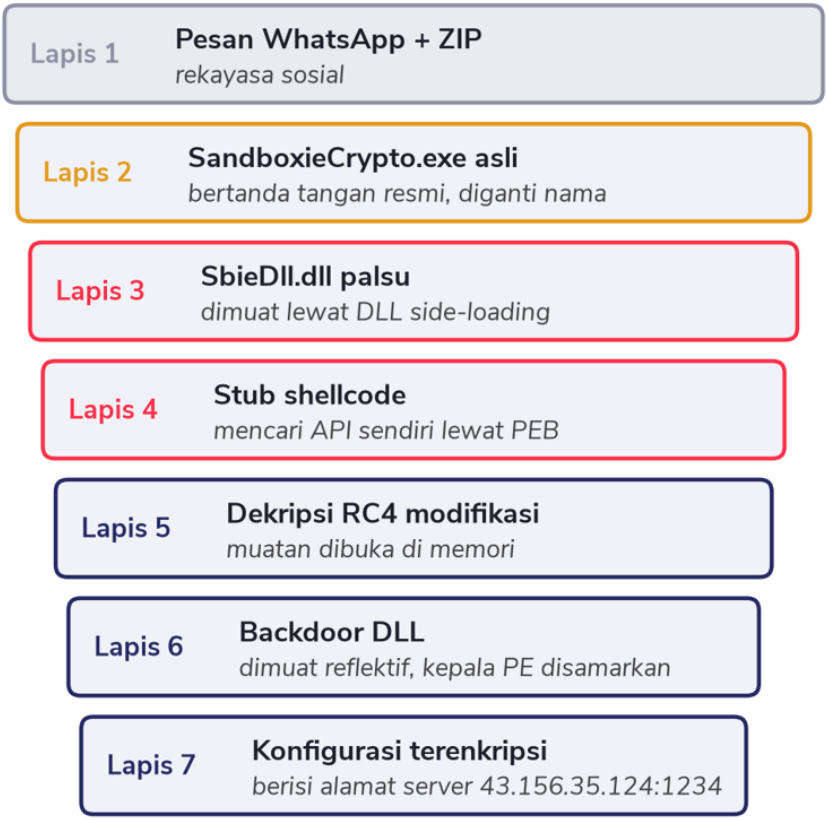
Bab ini membedah lampiran *Balance_by_Invoice_20261103.zip* yang diterima pada 16 November 2026. Isinya tiga file, yaitu program Sandboxie yang diganti nama, DLL palsu, dan file muatan .bin.



Gambar 3.1 Isi lampiran sampel November dan peran masing-masing file.

Sampel ini dirancang seperti bawang. Setiap lapis menutupi lapis berikutnya, dan setiap lapis punya tugas sendiri. Lapis luar bertugas menipu manusia. Lapis

tengah bertugas menipu sistem operasi dan antivirus. Lapis terdalam berisi program jahat yang sebenarnya beserta alamat tujuannya.



Gambar 3.2 Tujuh lapis penyamaran dalam sampel.

Seluruh pembedahan dilakukan secara statis di komputer macOS. Tidak ada satu pun file yang dijalankan. Setiap lapis dibuka dengan membaca byte, membongkar kode mesin menjadi instruksi yang bisa dibaca manusia, lalu meniru langkah dekripsinya sebagai pengolahan data biasa. Cara melakukannya diuraikan pada Bagian III.

3.1 Lapis 1, Pesan dan ZIP

Lapis pertama sudah dibahas pada Bab 1. Tugasnya membuat manusia mengekstrak ZIP di komputer Windows dan mengklik file yang tampak seperti faktur. Lapis ini sepenuhnya bergantung pada kelengahan manusia.

3.2 Lapis 2, Program Asli Bertanda Tangan

Informasi versi di dalam file *Ringkasan_Faktur_Pemasok_20261116.exe* menunjukkan identitas aslinya.

Tabel 3.1 Identitas asli file umpan

Atribut	Nilai
Nama asli	SandboxieCrypto.exe
Deskripsi	Sandboxie COM Services (CryptSvc)
Produk dan versi	Sandboxie 5.69.10
Perusahaan	Sandboxie-Plus.com
Waktu kompilasi	1 Oktober 2024
Tanda tangan	Sertifikat *Extended Validation* dari GlobalSign, cap waktu DigiCert
Jejak kompilasi	D:\a\Sandboxie\...\SandboxieCrypto.pdb

Program ini sah dan tidak berbahaya bila berada di folder instalasi Sandboxie. Yang dimanfaatkan penipu adalah perilakunya. Saat dijalankan, program ini mencari *SbieDll.dll* dan memuatnya. Windows mencari DLL mulai dari folder tempat program berada, sehingga DLL palsu di folder yang sama akan dimuat lebih dulu daripada DLL asli di tempat lain.

Mengapa penting

Tanda tangan digital hanya menjamin bahwa program tersebut asli buatan pengembangnya. Tanda tangan tidak menjamin bahwa semua file yang dimuat program itu juga asli. Celah inilah yang dimanfaatkan teknik *DLL side-loading*.

3.3 Lapis 3, DLL Palsu yang Menyamar

File *SbieDll.dll* berukuran 1,4 MB dan dikompilasi pada 25 Juli 2025. Informasi versinya mengaku sebagai *MD5 Edit Tools (Modified file fingerprint)* buatan pihak lain, lengkap dengan alamat situs dan email. Informasi ini palsu dan dipasang untuk mengecoh pemeriksaan sekilas. Manifes di dalamnya juga menunjukkan jejak lingkungan pemrograman *E-language*, bahasa pemrograman berbahasa Mandarin yang populer di kalangan pembuat perangkat lunak tertentu.

Di dalam DLL ini, nama-nama fungsi penting disembunyikan dengan pola *EncryptedFunctionPtr*, sehingga pemindaian teks biasa tidak menemukan perintah yang mencurigakan. Fungsi Windows yang masih terlihat sudah cukup menggambarkan tugasnya.

Tabel 3.2 Fungsi Windows yang terlihat pada DLL palsu

Fungsi	Kegunaan dalam serangan
CreateFileA, ReadFile	Membuka dan membaca file *SbieDll.bin*
VirtualAlloc	Menyiapkan area memori baru untuk kode
CreateThread	Menjalankan kode di area memori itu
LoadLibrary, GetProcAddress	Mencari fungsi Windows lain secara diam-diam
CryptCreateHash, CryptHashData	Menghitung sidik data, kemungkinan untuk memeriksa keutuhan muatan

Dengan kata lain, DLL palsu ini adalah pemuat (*loader*). Ia membaca isi *SbieDll.bin* ke memori, lalu menyerahkan kendali kepada kode yang ada di dalamnya.

3.4 Lapis 4, Stub Shellcode

File *SbieDll.bin* sekilas tampak seperti data acak. Program *file* bahkan salah mengenalinya sebagai kunci publik OpenPGP. Pemeriksaan byte demi byte memperlihatkan susunan yang rapi.



Gambar 3.3 Susunan isi file SbieDll.bin hasil pembedahan.

Empat byte pertama berisi angka 0xF98 atau 3.992, yaitu panjang potongan kode yang mengikutinya. Potongan kode ini disebut *stub shellcode*, kode mesin x64 mandiri yang tidak memerlukan bantuan sistem operasi untuk dijalankan. Instruksi pertamanya bisa langsung dibaca.

```
000004  mov qword ptr [rsp + 8], rcx
000009  sub rsp, 0x28
00000d  mov rcx, qword ptr [rsp + 0x30]
000012  call 0x240
```

Fungsi utama stub berada di posisi 0x240. Dari pembongkaran fungsi tersebut, urutan kerjanya bisa dipetakan sebagai berikut.

1. Membaca angka ukuran muatan yang tersimpan setelah stub.
2. Mencari alamat fungsi Windows sendiri dengan menelusuri struktur *Process Environment Block* (PEB) melalui register gs:[0x60]. Teknik ini membuat stub tidak perlu daftar impor yang bisa dicurigai.
3. Memanggil *ZwAllocateVirtualMemory* untuk menyiapkan memori. Nama fungsi ini disusun dari potongan angka heksadesimal agar tidak terbaca sebagai teks.
4. Menyalin muatan ke memori baru, lalu memanggil fungsi dekripsi di posisi 0xF3C.
5. Memuat hasil dekripsi sebagai program Windows langsung dari memori.
6. Menjalankan fungsi awal program tersebut dengan menyerahkan alamat seluruh isi *SbieDll.bin* sebagai bekal.

3.5 Lapis 5, RC4 yang Dimodifikasi

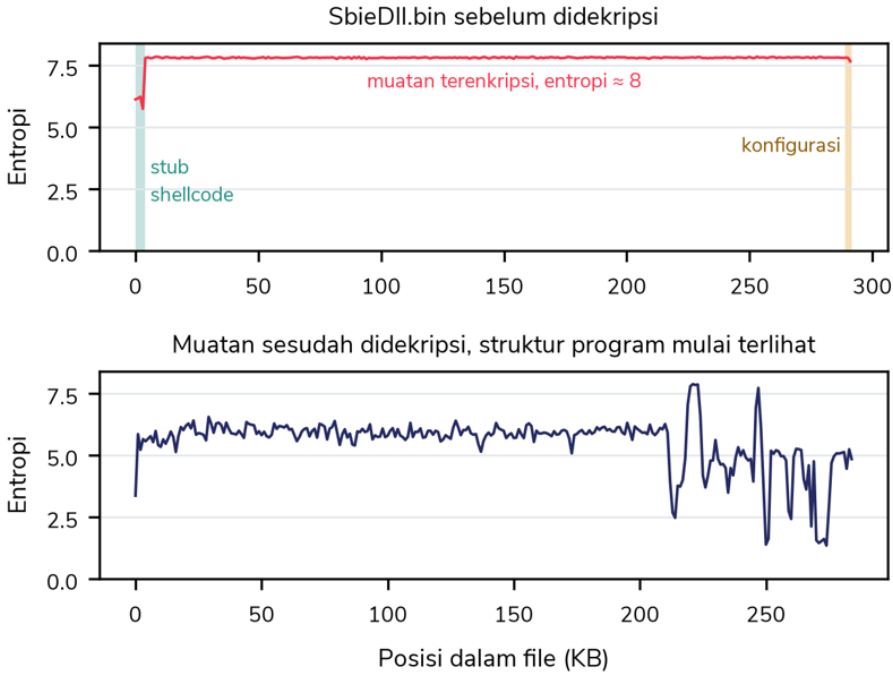
Fungsi dekripsi di posisi 0xF3C memanggil dua fungsi lain. Fungsi pertama di posisi 0xCC0 mengisi tabel 256 byte berisi angka 0 sampai 255, lalu mengacaknya berdasarkan kunci. Fungsi kedua di posisi 0xDD8 menghasilkan deretan byte acak dari tabel tersebut dan menggabungkannya dengan data menggunakan operasi XOR. Pola ini dikenal sebagai algoritma RC4.

Ada satu perbedaan kecil yang penting. Pada RC4 standar, byte kunci aliran diambil dari posisi S[(S[i] + S[j]) mod 256]. Pada sampel ini, byte kunci aliran dihitung sebagai S[i] XOR S[j]. Perubahan satu baris ini membuat alat dekripsi RC4 biasa gagal membuka muatan, dan membuat pola RC4 standar tidak dikenali oleh sebagian alat deteksi.

Tabel 3.3 Perbandingan RC4 standar dan versi sampel

Bagian	RC4 standar	Versi sampel
Pengacakan tabel	j = j + S[i] + kunci[i]	Sama
Pembaruan indeks	i = i + 1, j = j + S[i]	Sama
Byte kunci aliran	S[(S[i] + S[j]) mod 256]	S[i] XOR S[j]
Hasil uji pada sampel	Data tetap acak	Entropi turun, program terbaca

Kunci yang dipakai tersimpan di dalam file itu sendiri. Satu byte bernilai 210 menunjukkan panjang kunci, lalu 210 byte berikutnya adalah kunci. Sisanya, 291.629 byte, adalah muatan terenkripsi.



Gambar 3.4 Entropi SbieDll.bin per kilobyte sebelum dan sesudah didekripsi.

Grafik di atas memperlihatkan dampak dekripsi. Sebelum didekripsi, hampir seluruh file memiliki entropi mendekati 8 bit per byte, nilai maksimum yang menandakan data terenkripsi. Hanya potongan stub di awal yang lebih rendah. Sesudah didekripsi dengan RC4 versi sampel, entropi turun menjadi sekitar 6 dan polanya berubah-ubah, ciri khas program yang berisi kode, tabel, dan teks.

3.6 Lapis 6, Program yang Kepalanya Disamarkan

Hasil dekripsi ternyata tidak diawali tanda MZ seperti program Windows pada umumnya. Bagian awalnya berisi nol. Namun tabel bagian program seperti .text dan .rdata masih terlihat di posisi 365. Teka-teki ini terjawab dari kode stub. Pemuat membaca posisi kepala program dari alamat 0x3C pada awal buffer, dan awal buffer itu adalah kunci RC4 tadi. Artinya, kunci 210 byte tersebut sekaligus berfungsi sebagai kepala program.

Teknik ini membuat muatan tidak bisa dikenali sebagai program meskipun sudah didekripsi, kecuali bagian kunci dan hasil dekripsi digabung kembali. Setelah digabung, program dapat dibaca dengan normal.

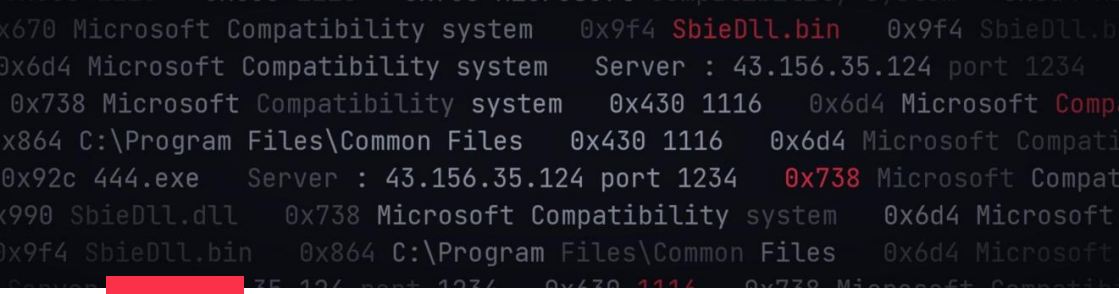
Tabel 3.4 Identitas program inti

Atribut	Nilai
Jenis	DLL Windows 64-bit
Waktu kompilasi	6 Maret 2026
Ukuran di memori	300 KB (0x4B000)
Bagian program	.text, .rdata, .data, .pdata, .reloc
Pustaka pihak ketiga	libwebsockets 3.2.3 dan zlib 1.2.13
Jumlah fungsi impor	259 fungsi dari 14 pustaka Windows
SHA-256 setelah direkonstruksi	Dicantumkan di Lampiran B

Stub menjalankan fungsi awal DLL (*DllMain*) lebih dulu, lalu mencari fungsi ekspor bernama *run*. Program inti ternyata tidak memiliki fungsi ekspor sama sekali, sehingga seluruh pekerjaan dimulai dari *DllMain*. Di dalam *DllMain*, program memeriksa dua hal, yaitu apakah ia baru dimuat dan apakah ia menerima bekal alamat isi *SbieDll.bin*. Bila keduanya terpenuhi, fungsi utama di posisi 0x6390 dijalankan.

3.7 Lapis 7, Konfigurasi Terenkripsi

Lapis terakhir menyimpan informasi paling berharga bagi penyelidik, yaitu alamat server tujuan. Pembahasannya cukup penting sehingga diberi bab tersendiri.



Bab 4

Ke Mana Perginya Sampel November

Program pengendali jarak jauh selalu membutuhkan alamat server untuk dihubungi. Alamat itu biasanya disimpan dalam konfigurasi yang dienkripsi. Pada sampel ini, konfigurasi berada di ujung file *SbieDll.bin*.

4.1 Menemukan Konfigurasi

Setelah muatan berakhir, masih ada 2.704 byte tersisa di ujung *SbieDll.bin*. Empat byte pertamanya bernilai 0xA88 atau 2.696, diikuti 2.696 byte data dengan entropi 7,93, lalu empat byte nol. Polanya jelas menunjukkan blok data terenkripsi dengan penanda panjang.

Fungsi pertama yang dipanggil program inti, di posisi 0xA4E8, menerima alamat isi *SbieDll.bin*. Fungsi ini menyalin stub, menyalin muatan, lalu menyalin tepat 0xA88 byte dari ujung file ke sebuah variabel global. Sesudah itu ia memanggil fungsi di posisi 0x8C6C.

4.2 Kunci yang Terlalu Sederhana

Fungsi di posisi 0x8C6C ternyata RC4 versi sampel yang sama. Perbedaannya ada pada kunci. Kuncinya ditulis langsung di dalam kode sebagai lima byte berurutan, yaitu 01 02 03 04 05.

```
008c8a  mov dword ptr [rbp + 0x30], 0x4030201
008c94  mov byte ptr [rbp + 0x34], 5
...
008d3b  mov cl, byte ptr [rsp + r8]
```

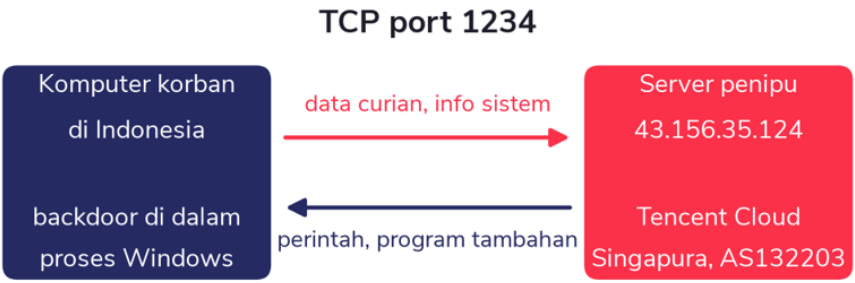
```
008d3f  xor cl, byte ptr [rsp + r9]
008d43  xor byte ptr [rsi], cl
```

Instruksi di posisi 0x8C8A dan 0x8C94 menyusun kunci 01 02 03 04 05. Tiga instruksi terakhir adalah inti RC4 versi sampel, yaitu S[i] XOR S[j] yang kemudian di-XOR dengan data. Dengan kunci ini, konfigurasi dapat dibuka sepenuhnya.

4.3 Isi Konfigurasi

Tabel 4.1 Isi konfigurasi setelah didekripsi

Posisi	Nilai	Makna
0x000	43.156.35.124	Alamat server utama
0x100	1234	Port tujuan
0x102	TCP	Protokol
0x10C, 0x218, 0x324	43.156.35.124 : 1234 TCP	Tiga slot cadangan, semuanya diisi alamat yang sama
0x430	1116	Tanda kampanye, sesuai tanggal 16 November
0x670 s.d. 0x738	Microsoft Compatibility system	Nama penyamaran agar terlihat seperti komponen Windows
0x864	C:\Program Files\Common Files	Folder tujuan penyalinan
0x92C	444.exe	Nama baru untuk program umpan
0x990, 0x9F4	SbieDll.dll, SbieDll.bin	File pendamping yang ikut disalin



Tanda kampanye "1116", nama penyamaran "Microsoft Compatibility system"

Gambar 4.1 Arah komunikasi antara komputer korban dan server penipu.

4.4 Siapa Pemilik Alamat Itu

Pencarian data WHOIS untuk alamat 43.156.35.124 menunjukkan bahwa alamat tersebut termasuk blok 43.156.0.0/16 milik Aceville Pte. Ltd. di Singapura, dengan nomor sistem otonom AS132203. Aceville adalah entitas yang mengoperasikan layanan Tencent Cloud di luar Tiongkok. Kontak pengaduan penyalahgunaan untuk blok ini adalah abuse@tencent.com.

Temuan ini perlu dibaca dengan hati-hati. Alamat tersebut adalah server awan yang disewa. Siapa pun di mana pun bisa menyewanya dengan biaya murah, sehingga lokasi server tidak menunjukkan kewarganegaraan atau lokasi pelaku. Pada saat buku ini disusun, alamat tersebut belum tercatat di basis data ancaman publik, yang menandakan kampanye masih baru.

⚠ Catatan penting

Pembedahan ini tidak pernah menghubungi server 43.156.35.124. Menghubungi server penipu dapat memberi tahu pelaku bahwa sampelnya sedang diselidiki, dan dapat membuat alamat IP penyelidik tercatat di sisi pelaku.

4.5 Apa yang Terjadi Setelah Terhubung

Kehadiran pustaka libwebsockets dan zlib menunjukkan bahwa komunikasi dilakukan dengan protokol WebSocket di atas TCP, dan data dikompresi sebelum dikirim. Pola ini lazim pada program pengendali jarak jauh modern. Komputer korban membuka koneksi ke server, mengirim informasi sistem, lalu menunggu perintah. Perintah bisa berupa pengambilan file, pengambilan layar, eksekusi perintah, atau pengiriman program tambahan yang lebih spesifik, misalnya pencuri kata sandi.

Bab 5

Sampel Oktober, Muatan di Dalam Tanda Tangan

Lampiran *Ringkasan_20261019_Saldo_Tagihan.zip* diterima pada 21 Oktober 2026 dari nomor +91. Naskah pesannya sama dengan sampel November, tetapi isi lampirannya dibuat dengan cara yang sama sekali berbeda. Perbedaan inilah yang membuktikan bahwa yang perlu dipahami adalah modelnya, karena bahannya terus diganti.



Gambar 5.1 Rantai pemuatan sampel Oktober.

5.1 Dua File dengan Tanda Tangan Resmi

Tabel 5.1 Isi lampiran sampel Oktober

Nama file	Ukuran	Hasil pemeriksaan
2026Pmksnflacker.exe	843 KB	Program Windows 64-bit bertanda tangan sah atas nama NetEase (Hangzhou) Network Co., Ltd., dikompilasi 11 Agustus 2026. Tidak memiliki informasi versi.

Nama file	Ukuran	Hasil pemeriksaan
CrashHunter_PC3.dll	3,2 MB	Pustaka pelaporan galat *CrashHunter* (Game SDK 1.7.0.0) yang aslinya juga milik NetEase, dikompilasi 11 April 2025. File ini sudah ditambal sehingga tanda tangannya tidak lagi sah.

Program *2026Pmksnflacker.exe* memanggil fungsi *getInstance* dari *CrashHunter_PC3.dll*, sehingga Windows otomatis memuat DLL tersebut dari folder yang sama. Ini adalah *DLL side-loading* yang sama seperti sampel November, dengan program sah dari pengembang yang berbeda. NetEase tidak terlibat dalam serangan ini. Programnya dipakai tanpa izin.

5.2 DLL Asli yang Ditambal

Berbeda dari sampel November yang memakai DLL palsu, sampel Oktober memakai DLL asli yang dimodifikasi. Tanda pertamanya terlihat pada daftar bagian program. Selain bagian standar seperti .text dan .rdata, ada satu bagian tambahan berukuran 2,5 KB dengan nama acak **.yls3k4v**.

Titik masuk DLL juga dipindahkan ke potongan kode kecil yang disisipkan di ruang kosong akhir bagian .text. Potongan ini menjalankan dua hal berurutan. Pertama, ia memanggil fungsi awal DLL yang asli agar *CrashHunter* tetap bekerja normal. Kedua, ia memanggil kode di bagian .yls3k4v.

```
228dbf  lea rax, [rip - 0x56bca]   ; DllMain asli
228dc6  call rax
228dd4  lea rax, [rip + 0x119225]  ; 0x342000, .yls3k4v
228ddb  call rax
```

Dengan cara ini program tetap tampak bekerja seperti biasa, sementara kode tambahan berjalan diam-diam di belakangnya.

5.3 Membaca Dirinya Sendiri

Kode di bagian .yls3k4v menelusuri PEB untuk mencari fungsi Windows, sama seperti stub sampel November. Fungsi-fungsi itu dicari dengan membandingkan nilai *hash* namanya, misalnya 0x7C0017A5, sehingga nama fungsi tidak pernah muncul sebagai teks. Setelah itu kode tersebut mengambil jalur file program induknya sendiri, lalu menyusun sebuah kunci dari potongan angka di memori.

```
3420a6  mov dword ptr [rsp + 0x20], 0x456b7441   ; AtkE
3420b2  mov dword ptr [rsp + 0x24], 0x626d4578   ; xEmb
3420bf  mov dword ptr [rsp + 0x28], 0x654b6465   ; edKe
3420c7  mov dword ptr [rsp + 0x2c], 0x32303279   ; y202
3420cf  mov byte ptr [rsp + 0x30], 0x36           ; 6
```

Bila dibaca dalam urutan *little-endian*, kelima nilai itu membentuk teks **AtkExEmbedKey2026**. Nama kunci ini sudah memberi petunjuk bahwa ada sesuatu yang disematkan (*embed*) di suatu tempat.

5.4 Muatan di Area Tanda Tangan Digital

Kode tersebut membuka file *2026Pmksnflacker.exe* miliknya sendiri, membaca kepala program, lalu melompat ke direktori keamanan, yaitu area tempat tanda tangan digital Authenticode disimpan. Di dalam area berukuran 41.032 byte itu, ia mencari penanda **CA FE BA BE**. Data setelah penanda tersebut didekripsi dengan RC4 standar memakai kunci *AtkExEmbedKey2026*, diubah menjadi kode yang bisa dijalankan, lalu dijalankan.

⚠ Mengapa cerdik

Saat memeriksa keaslian tanda tangan, Windows sengaja tidak menghitung isi area tanda tangan itu sendiri. Akibatnya data tambahan yang diselipkan di sana tidak membatalkan tanda tangan. Program tetap tampil sebagai program sah buatan NetEase, padahal membawa muatan jahat di dalamnya.

5.5 Tahap Dua, Pengunduh

Hasil dekripsi berupa *shellcode* x64 berukuran 30.272 byte dengan entropi 5,41. Di dalamnya terdapat penanda teks **SCURLTPL**, diikuti deretan karakter acak yang diakhiri banyak huruf Z. Huruf Z berulang adalah petunjuk bahwa teks tersebut di-XOR dengan nilai 0x5A, karena byte nol yang di-XOR dengan 0x5A akan menjadi huruf Z.

Setelah di-XOR dengan 0x5A, teks tersebut berubah menjadi alamat unduhan berikut.

```
https://blasidha.eu.cc/files/quoweu.eu.cc.bin
```

Artinya, tahap dua bertugas sebagai pengunduh. Ia mengambil tahap tiga dari alamat tersebut, lalu menjalankannya. Isi tahap tiga tidak ikut diperiksa, karena mengunduhnya berarti menghubungi server pelaku.

5.6 Ke Mana Perginya

Domain eu.cc adalah layanan subdomain gratis. Siapa pun dapat membuat subdomain di bawahnya dalam hitungan menit dan membuangnya kapan saja, sehingga layanan seperti ini kerap dipakai untuk menampung program jahat. Basis data URLhaus milik abuse.ch, misalnya, telah mencatat subdomain eu.cc lain sebagai penyebar program jahat. Pada saat buku ini disusun, subdomain *blasidha.eu.cc* belum tercatat di sumber publik.

5.7 Dua Sampel, Satu Model

Tabel 5.2 Perbandingan teknis dua sampel

Unsur	Sampel Oktober	Sampel November
Program yang ditunggangi	Program NetEase bertanda tangan sah	SandboxieCrypto.exe bertanda tangan sah
Pembuka	DLL asli CrashHunter yang ditambal	DLL palsu bernama SbieDll.dll

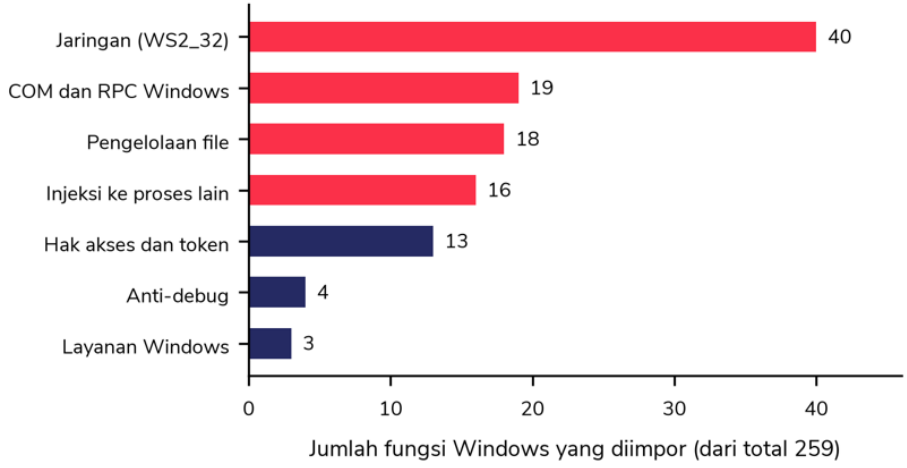
Unsur	Sampel Oktober	Sampel November
Lokasi muatan	Area tanda tangan digital program	File SbieDll.bin terpisah
Enkripsi	RC4 standar, kunci AtkExEmbedKey2026	RC4 modifikasi, kunci 210 byte
Pencarian fungsi Windows	Penelusuran PEB dan *hash* nama	Penelusuran PEB
Tujuan	Mengunduh dari blasidha.eu.cc	Terhubung ke 43.156.35.124:1234

Dua sampel ini dirakit dengan alat, kunci, dan program tunggangan yang berbeda. Keduanya tetap mengikuti enam tahap yang sama, yaitu umpan bertema keuangan, arsip pembawa, program sah sebagai pembuka, muatan tersembunyi yang dijalankan di memori, koneksi ke server pelaku, lalu panen.

Bab 6

Kemampuan Program Inti

Bab ini membahas program inti dari sampel November yang berhasil dibuka utuh. Muatan akhir sampel Oktober diunduh dari internet sehingga tidak ikut terbawa di dalam lampiran, tetapi rantai pemuatnya dirancang untuk mengantarkan program sejenis. Jenis pasti program jahat paling akurat ditentukan dengan menjalankannya di lingkungan terkendali. Tanpa menjalankannya pun, daftar fungsi Windows yang diimpor dan teks yang disembunyikan di dalamnya sudah memberi gambaran yang kuat tentang kemampuannya.



Gambar 6.1 Sebaran fungsi Windows yang diimpor program inti menurut kelompok kemampuan.

6.1 Membaca Daftar Impor

Tabel 6.1 Kelompok fungsi impor dan artinya

Kelompok	Contoh fungsi	Artinya
Jaringan	WSAStartup, connect, send, recv, listen, accept, getaddrinfo	Berkomunikasi dengan server, bahkan bisa membuka port untuk menerima koneksi
Injeksi proses	VirtualAllocEx, WriteProcessMemory, CreateRemoteThread, SetThreadContext	Menyuntikkan kode ke proses Windows lain agar tersembunyi
Hak akses	DuplicateTokenEx, CreateProcessWithTokenW, WTSQueryUserToken	Meminjam hak akses pengguna lain atau hak sistem
Layanan	OpenSCManagerW, EnumServicesStatusExW	Memeriksa layanan Windows yang berjalan
COM dan RPC	CoCreateInstance, RpcBindingFromStringBindingW	Mengendalikan komponen Windows, misalnya penjadwal tugas
Anti-debug	NtRemoveProcessDebug, DbgUiSetThreadDebugObject	Melepaskan diri dari alat analisis
Lain-lain	SetThreadExecutionState, DeviceIoControl	Mencegah komputer tidur, berkomunikasi dengan driver

6.2 Teks yang Disembunyikan

Program inti menyembunyikan teks penting dengan teknik *stack string* yang di-XOR. Setiap teks disimpan sebagai dua kelompok angka 64-bit yang tidak bermakna bila dibaca sendiri-sendiri. Saat dibutuhkan, kedua kelompok di-XOR untuk menghasilkan teks aslinya. Empat teks yang ditemukan di fungsi utama sebagai berikut.

Tabel 6.2 Teks yang dibuka dari fungsi utama

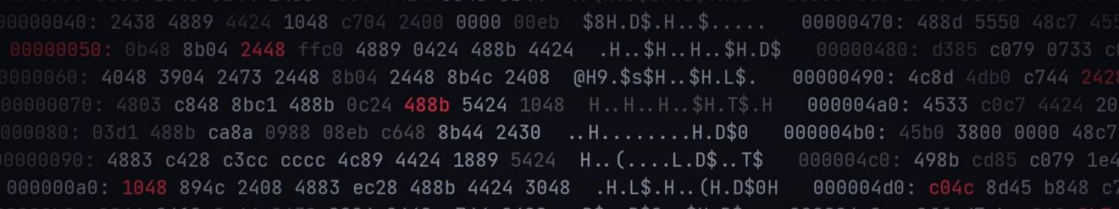
Teks	Dugaan kegunaan
svchost.exe	Memeriksa apakah program sedang berjalan di dalam proses layanan Windows
Schedule	Nama layanan Penjadwal Tugas Windows, lazim dipakai untuk bertahan setelah komputer dinyalakan ulang
vssvc.exe	Nama proses layanan *Volume Shadow Copy*, kemungkinan dipakai sebagai proses inang penyamaran
%s\%s	Pola penyusunan jalur folder dan nama file

6.3 Perilaku yang Bisa Disimpulkan

Dengan menggabungkan konfigurasi, daftar impor, dan teks tersembunyi, perilaku program inti dapat disimpulkan sebagai berikut.

1. Menyalin dirinya ke folder *C:\Program Files\Common Files* dengan nama yang tidak mencolok.
2. Mendaftarkan diri agar berjalan kembali setiap kali komputer menyala, dengan nama yang mirip komponen resmi Microsoft.
3. Berpindah ke dalam proses Windows yang sah, misalnya *svchost.exe*, agar tidak terlihat sebagai program asing di Task Manager.
4. Menghubungi 43.156.35.124 port 1234 dan menunggu perintah.
5. Menjalankan perintah pelaku, termasuk mengunduh dan menjalankan program tambahan dengan hak akses yang lebih tinggi.

Pola seperti ini, yaitu program bertanda tangan sah, DLL palsu, muatan terenkripsi yang dimuat di memori, dan program pengendali jarak jauh berbasis WebSocket, telah berulang kali dilaporkan oleh peneliti keamanan dalam kampanye yang menyasar pengguna di Asia. Buku ini sengaja tidak menempelkan nama kelompok atau keluarga program jahat tertentu, karena penamaan seperti itu membutuhkan pembanding yang lebih luas daripada satu sampel.



Bab 7

Apa yang Akan Diambil

Program pengendali jarak jauh memberi pelaku kendali yang hampir setara dengan duduk di depan komputer korban. Apa pun yang bisa dilakukan pengguna di komputer itu, pelaku juga bisa melakukannya, sering kali tanpa terlihat.



Gambar 7.1 Data dan akses yang umumnya diincar program jahat jenis ini.

7.1 Data yang Diincar

Tabel 7.1 Data yang diincar dan akibatnya

Yang diambil	Akibat bagi korban
Kata sandi di browser	Semua akun yang sandinya disimpan di Chrome, Edge, atau Firefox bisa dibuka pelaku.
Cookie sesi	Pelaku bisa masuk ke email dan media sosial korban tanpa kata sandi dan tanpa kode OTP, karena sesi yang sudah masuk ikut dicuri.
Sesi WhatsApp Web dan Telegram	Pelaku bisa membaca percakapan dan mengirim pesan atas nama korban kepada rekan kerja dan pelanggan.

Yang diambil	Akibat bagi korban
***Internet banking* dan aplikasi keuangan**	Pelaku bisa mengintip saldo, mempelajari cara korban bertransaksi, atau menunggu korban login lalu mengambil alih.
Dokumen kerja	Faktur, kontrak, data pelanggan, dan data pemasok bisa disalin untuk penipuan berikutnya atau dijual.
Ketikan dan layar	Pelaku bisa merekam semua yang diketik dan mengambil gambar layar secara berkala.
Dompet kripto	Isi dompet kripto di komputer bisa dikuras.
Jaringan kantor	Komputer korban menjadi pintu masuk ke komputer dan server lain di jaringan yang sama.

7.2 Ujung yang Paling Merugikan

Kerugian terbesar dari modus seperti ini muncul ketika pelaku sudah memahami alur tagihan perusahaan. Dari email dan dokumen yang dicuri, pelaku tahu siapa pemasok perusahaan, berapa nilai tagihannya, dan kapan jatuh temponya.

Pelaku kemudian mengirim email atau pesan WhatsApp yang terlihat berasal dari pemasok asli, kadang memakai akun yang sudah diretas, dengan kabar bahwa rekening tujuan pembayaran berubah. Bila bagian keuangan tidak memeriksa ulang, pembayaran tagihan yang sah akan masuk ke rekening pelaku. Modus ini dikenal sebagai *business email compromise* dan kerugiannya bisa mencapai ratusan juta rupiah dalam satu transaksi.

Ingat

Setelah komputer dikuasai, pelaku tidak perlu terburu-buru. Mereka bisa diam berminggu-minggu, membaca email dan percakapan, lalu bergerak pada saat ada pembayaran besar.

7.3 Korban Berikutnya

Akun WhatsApp dan email korban yang sudah dikuasai menjadi alat untuk menyebarkan file yang sama. Kali ini pesannya datang dari nomor atau alamat yang dikenal, sehingga penerima lebih mudah percaya. Satu klik bisa menjalar ke rekan kerja, pemasok, dan pelanggan.

7.4 Pemetaan ke MITRE ATT&CK

Bagi tim keamanan, perilaku sampel dapat dipetakan ke kerangka MITRE ATT&CK agar mudah dibandingkan dengan laporan lain dan dipakai untuk menyusun aturan deteksi.

Tabel 7.2 Pemetaan teknik ke MITRE ATT&CK

ID	Teknik	Bukti pada sampel
T1566	Phishing	Pesan WhatsApp berisi lampiran faktur palsu
T1204.002	User Execution, Malicious File	Korban harus mengklik file .exe
T1036	Masquerading	Nama file faktur, nama layanan mirip Microsoft
T1574.002	DLL Side-Loading	SandboxieCrypto.exe memuat SbieDll.dll palsu
T1027	Obfuscated Files or Information	RC4 modifikasi, stack string XOR, fungsi terenkripsi
T1620	Reflective Code Loading	Muatan dimuat langsung dari memori
T1055	Process Injection	WriteProcessMemory, CreateRemoteThread, SetThreadContext
T1134	Access Token Manipulation	DuplicateTokenEx, CreateProcessWithTokenW
T1053.005	Scheduled Task	Teks *Schedule*, pemakaian COM dan RPC

ID	Teknik	Bukti pada sampel
T1071	Application Layer Protocol	WebSocket melalui libwebsockets
T1571	Non-Standard Port	Port 1234
T1027.009	Embedded Payloads	Muatan sampel Oktober diselipkan di area tanda tangan digital
T1036.001	Invalid Code Signature	CrashHunter_PC3.dll ditambal sehingga tanda tangannya tidak sah
T1105	Ingress Tool Transfer	Tahap dua sampel Oktober mengunduh tahap tiga dari blasidha.eu.cc

Bagian III

MENYELIDIKI SENDIRI DENGAN AMAN

Langkah investigasi yang bisa diulang sendiri secara aman, lengkap dengan skrip.

> **“Security is a process, not a product.”**
>
> *Keamanan adalah sebuah proses. Ia tidak bisa dibeli sebagai produk.*
>
> Bruce Schneier, Crypto-Gram, 15 Mei 2000

Bab 8

Prinsip dan Persiapan

Bagian ini menunjukkan cara mengulang pembedahan pada Bagian II. Langkah-langkahnya dapat dipakai oleh tim TI perusahaan, mahasiswa, atau siapa pun yang ingin memahami cara kerja program jahat tanpa membahayakan diri sendiri. Semua contoh perintah sudah diuji pada sampel yang dibahas di buku ini.

8.1 Analisis Statis dan Dinamis

Ada dua pendekatan utama dalam analisis program jahat. Analisis statis memeriksa file tanpa menjalankannya. Analisis dinamis menjalankan file di lingkungan terkendali lalu mengamati perilakunya.

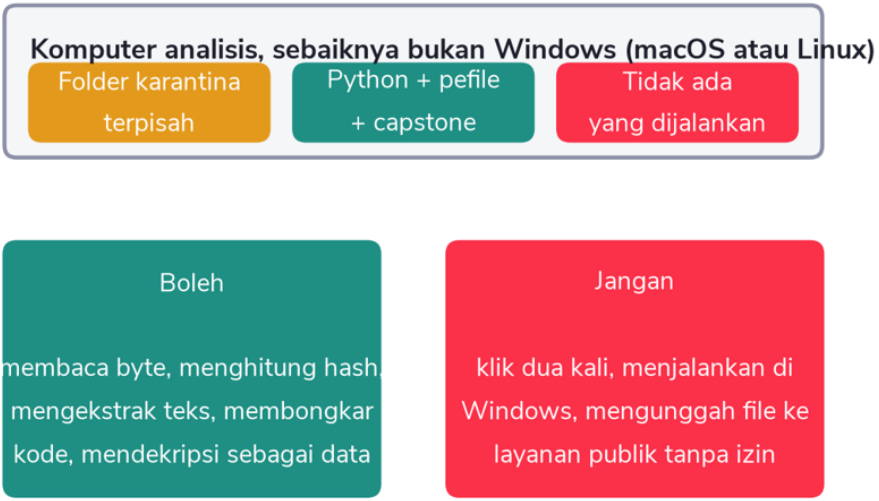
Tabel 8.1 Perbandingan analisis statis dan dinamis

Aspek	Statis	Dinamis
Cara	Membaca byte, teks, struktur, dan kode mesin	Menjalankan sampel di mesin virtual atau *sandbox*
Risiko	Sangat rendah bila tidak ada yang dijalankan	Lebih tinggi, sampel benar-benar aktif
Kelebihan	Aman, bisa membuka konfigurasi dan kunci	Cepat melihat perilaku nyata
Kekurangan	Butuh waktu dan pemahaman kode mesin	Sampel bisa mendeteksi *sandbox* dan diam

Aspek	Statis	Dinamis
Kontak dengan pelaku	Tidak ada	Sampel bisa menghubungi server pelaku

Buku ini sepenuhnya memakai analisis statis. Pendekatan ini dipilih karena aman dijalankan di komputer biasa dan tidak memberi tahu pelaku bahwa sampelnya sedang diselidiki.

8.2 Aturan Keselamatan



Gambar 8.1 Prinsip lingkungan analisis yang aman.

1. Lakukan analisis di komputer yang tidak bisa menjalankan program sasaran. Untuk program Windows, gunakan macOS atau Linux.
2. Simpan sampel di folder karantina terpisah. Jangan simpan skrip analisis di folder yang sama dengan sampel.

3. Jangan pernah mengklik dua kali file sampel, bahkan di komputer yang dianggap aman.
4. Beri akhiran khusus, misalnya .mal, pada setiap file hasil dekripsi agar tidak dijalankan secara tidak sengaja.
5. Jangan mengunggah sampel ke layanan analisis publik bila sampel mungkin berisi data perusahaan. Cari dengan nilai *hash* saja.
6. Jangan menghubungi alamat server yang ditemukan. Cukup catat dan laporkan.
7. Setelah selesai, hapus sampel atau simpan dalam arsip ZIP berkata sandi seperti kebiasaan para peneliti, misalnya dengan kata sandi *infected*.

8.3 Alat yang Dibutuhkan

Tabel 8.2 Alat analisis yang dipakai

Alat	Kegunaan
Terminal macOS atau Linux	Menjalankan perintah *file*, *shasum*, *strings*, *xxd*, dan *whois*
Python 3	Menjalankan skrip analisis
pefile	Pustaka Python untuk membaca struktur program Windows
capstone	Pustaka Python untuk membongkar kode mesin menjadi instruksi
Ghidra (opsional)	Alat bongkar kode gratis dari NSA untuk analisis yang lebih nyaman

Pustaka Python sebaiknya dipasang dalam lingkungan virtual agar terpisah dari sistem.

```
python3 -m venv ~/lab-bedah
~/lab-bedah/bin/pip install pefile capstone
```

Bab 9

Langkah Investigasi

Bab ini menguraikan dua belas langkah investigasi, dari mengenali jenis file sampai menemukan alamat server pelaku. Langkah 1 sampai 10 memakai sampel November, sedangkan Langkah 11 dan 12 memakai sampel Oktober. Setiap langkah disertai perintah atau skrip beserta hasilnya pada sampel.

9.1 Langkah 1, Mengenali Jenis File

Nama dan ekstensi file bisa dipalsukan. Perintah *file* membaca isi byte di awal file untuk menentukan jenis sebenarnya.

```
$ file *
Ringkasan_Faktur_Pemasok_20261116.exe:
    PE32+ executable (GUI) x86-64, for MS Windows
SbieDll.bin:  OpenPGP Public Key
SbieDll.dll:  PE32+ executable (DLL) (GUI) x86-64
```

Hasil ini langsung menjawab pertanyaan pertama. Lampiran yang mengaku faktur ternyata berisi program Windows 64-bit dan pustaka DLL. File .bin dikenali keliru sebagai kunci OpenPGP, tanda bahwa isinya acak dan tidak berpola.

9.2 Langkah 2, Mencatat Sidik Digital

Nilai *hash* SHA-256 adalah sidik unik setiap file. Nilai ini dipakai untuk mencari laporan di basis data ancaman tanpa perlu mengunggah filenya.

```
$ shasum -a 256 *
185b7a48...89ba37  Ringkasan_Faktur_Pemasok_20261116.exe
97a048f3...40eb09  SbieDll.bin
c7398663...351c96  SbieDll.dll
```

Di Windows, perintah setara adalah *Get-FileHash* di PowerShell. Nilai lengkapnya tercantum di Lampiran B. Pada saat diperiksa, ketiga nilai ini belum ditemukan di sumber publik, tanda bahwa sampel masih baru.

9.3 Langkah 3, Membaca Identitas Program

Program Windows menyimpan informasi versi, nama perusahaan, dan waktu kompilasi. Informasi ini bisa dibaca dengan pustaka pefile.

02_versi.py

```
import sys, pefile, datetime

pe = pefile.PE(sys.argv[1])
ts = pe.FILE_HEADER.TimeDateStamp
print("Dikompilasi:",
      datetime.datetime.fromtimestamp(ts, datetime.UTC))
for info in getattr(pe, "FileInfo", []):
    for blok in info:
        for tabel in getattr(blok, "StringTable", []):
            for k, v in tabel.entries.items():
                print(f"  {k.decode()}: {v.decode()}")
```

Hasilnya pada file umpan menunjukkan *OriginalFilename* SandboxieCrypto.exe, perusahaan Sandboxie-Plus.com, versi 5.69.10, dikompilasi 1 Oktober 2024. Pada *SbieDll.dll*, hasilnya mengaku *MD5 Edit Tools* yang dikompilasi 25 Juli 2025. Ketidakcocokan antara nama file, nama perusahaan, dan isi adalah tanda bahaya yang kuat.

9.4 Langkah 4, Mencari Teks yang Terbaca

Perintah *strings* menampilkan deretan huruf yang terbaca di dalam file. Gabungkan dengan *grep* untuk menyaring kata kunci.

```
$ strings -a SbieDll.dll | grep -E \
    "VirtualAlloc|CreateThread|ReadFile|\.pdb"
CreateThread
ReadFile
VirtualAlloc
$ strings -a Ringkasan*.exe | grep pdb
D:\a\Sandboxie\...\SandboxieCrypto.pdb
```

Kombinasi *ReadFile*, *VirtualAlloc*, dan *CreateThread* pada sebuah DLL yang mengaku alat pengubah MD5 adalah pola klasik pemuat. Ia membaca sesuatu dari file, menyiapkan memori, lalu menjalankannya.

9.5 Langkah 5, Mengukur Entropi

Entropi mengukur seberapa acak isi sebuah file. Nilainya antara 0 dan 8 bit per byte. Teks biasa berkisar 4 sampai 5, program biasa 5 sampai 6,5, sedangkan data terenkripsi atau terkompresi mendekati 8.

01_entropi.py

```
import sys, math, collections

def entropi(data):
    hitung = collections.Counter(data)
    n = len(data)
    return -sum(c / n * math.log2(c / n)
                for c in hitung.values())

for nama in sys.argv[1:]:
    isi = open(nama, "rb").read()
    print(f"{nama}: {len(isi)} byte, "
          f"entropi {entropi(isi):.2f}")
```

```
Ringkasan_Faktur_Pemasok_20261116.exe: entropi 6.32
SbieDll.bin: 298544 byte, entropi 8.00
SbieDll.dll: 1473480 byte, entropi 5.68
```

Nilai 8,00 pada *SbieDll.bin* memastikan file tersebut terenkripsi. Di sinilah muatan utama disembunyikan.

9.6 Langkah 6, Memetakan Struktur File .bin

Perintah *xxd* menampilkan isi file dalam bentuk heksadesimal. Enam belas byte pertama memperlihatkan pola yang menarik.

```
$ xxd -l 32 SbieDll.bin
00000000: 980f 0000 4889 4c24 0848 83ec 2848 8b4c
00000010: 2430 e829 0200 0090 4883 c428 c3cc cccc
```

Empat byte pertama, 98 0F 00 00, adalah angka 0xF98 dalam urutan *little-endian*. Byte berikutnya, 48 89 4C 24 08, adalah pembuka fungsi x64 yang sangat umum. Artinya file ini diawali angka panjang lalu kode mesin. Skrip berikut membaca seluruh penanda panjang di dalam file.

03_kepala_bin.py

```
import sys, struct

b = open(sys.argv[1], "rb").read()
stub = struct.unpack("<I", b[0:4])[0]
muat = struct.unpack("<I", b[4 + stub:8 + stub])[0]
awal = 8 + stub
kl = b[awal]
konf = awal + muat
panjang = struct.unpack("<I", b[konf:konf + 4])[0]
print("panjang stub     :", stub)
print("panjang muatan   :", muat)
print("panjang kunci    :", kl)
print("posisi konfigurasi:", konf + 4)
print("panjang konfig   :", panjang)
```

```
panjang stub     : 3992
panjang muatan   : 291840
panjang kunci    : 210
posisi konfigurasi: 295844
panjang konfig   : 2696
```

9.7 Langkah 7, Membongkar Kode Stub

Pustaka capstone mengubah byte kode mesin menjadi instruksi yang bisa dibaca. Skrip berikut membongkar sejumlah instruksi mulai dari posisi tertentu.

04_bongkar.py

```
import sys, capstone

b = open(sys.argv[1], "rb").read()
mulai = int(sys.argv[2], 0)
jumlah = int(sys.argv[3])
md = capstone.Cs(capstone.CS_ARCH_X86,
                 capstone.CS_MODE_64)
kode = b[mulai:mulai + 0x1000]
for k, ins in enumerate(md.disasm(kode, mulai)):
    print(f"{ins.address:06x}  {ins.mnemonic} {ins.op_str}")
    if k + 1 >= jumlah:
        break
```

Perintah *python 04_bongkar.py SbieDll.bin 0x4 4* menampilkan empat instruksi pertama stub. Dengan mengikuti setiap instruksi *call*, fungsi demi fungsi bisa ditelusuri. Pola yang perlu dicari dalam stub semacam ini antara lain akses ke gs:[0x60] untuk penelusuran PEB, perulangan 256 kali untuk penyusunan tabel RC4, dan operasi XOR di dalam perulangan untuk dekripsi.

<> Tips membaca kode mesin

Tidak perlu memahami setiap instruksi. Fokuslah pada angka konstan yang mencolok, misalnya 0x100 yang menandakan tabel 256 byte, pemanggilan fungsi, dan instruksi XOR di dalam perulangan. Ketiganya sering langsung menunjukkan algoritma yang dipakai.

9.8 Langkah 8, Mendekripsi Muatan

Setelah algoritma diketahui, dekripsi bisa ditiru dengan Python. Fungsi berikut adalah RC4 versi sampel. Bedanya dengan RC4 standar hanya pada baris terakhir di dalam perulangan.

rc4x.py

```
def rc4x(kunci, data):
    """RC4 versi sampel: byte kunci = S[i] XOR S[j]."""
    S = list(range(256))
```

```
    j = 0
    for i in range(256):
        j = (j + S[i] + kunci[i % len(kunci)]) & 255
        S[i], S[j] = S[j], S[i]
    i = j = 0
    hasil = bytearray()
    for c in data:
        i = (i + 1) & 255
        j = (j + S[i]) & 255
        S[i], S[j] = S[j], S[i]
        hasil.append(c ^ S[i] ^ S[j])
    return bytes(hasil)
```

Skrip berikut memakai fungsi tersebut untuk membuka muatan, lalu memulihkan tanda *MZ* dan *PE* yang sengaja dihapus agar hasilnya bisa dibaca pefile.

05_buka_muatan.py

```
import sys, struct
from rc4x import rc4x

b = open(sys.argv[1], "rb").read()
stub = struct.unpack("<I", b[0:4])[0]
muat = struct.unpack("<I", b[4 + stub:8 + stub])[0]
p = b[8 + stub:8 + stub + muat]
kl = p[0]
kunci, data = p[1:1 + kl], p[1 + kl:]
buka = rc4x(kunci, data)
citra = bytearray(p[:1 + kl] + buka)
lf = struct.unpack("<I", citra[0x3C:0x40])[0]
citra[0:2] = b"MZ"
citra[lf:lf + 4] = b"PE\0\0"
open(sys.argv[2], "wb").write(citra)
print("e_lfanew:", hex(lf), "ukuran:", len(citra))
```

⚠ Keselamatan

File hasil langkah ini adalah program jahat yang utuh. Simpan dengan akhiran .mal, jangan dipindahkan ke komputer Windows, dan hapus setelah analisis selesai.

9.9 Langkah 9, Membaca Kemampuan Program Inti

Daftar fungsi impor pada program hasil dekripsi menunjukkan kemampuannya, seperti yang dibahas pada Bab 6.

06_impor.py

```
import sys, pefile

pe = pefile.PE(sys.argv[1])
print("Mesin :", hex(pe.FILE_HEADER.Machine))
print("Jenis :", "DLL" if pe.is_dll() else "EXE")
for s in pe.sections:
    print("Bagian:", s.Name.rstrip(b"\0").decode())
for d in pe.DIRECTORY_ENTRY_IMPORT:
    nama = [i.name.decode() for i in d.imports if i.name]
    print(d.dll.decode(), len(nama), nama[:5])
```

Teks yang disembunyikan dengan *stack string* XOR dapat dibuka dengan cara yang sama seperti program melakukannya. Angka-angkanya diambil dari instruksi *movabs* pada hasil bongkar fungsi utama.

07_xor_teks.py

```
import struct

def buka(data, kunci):
    a = b"".join(struct.pack("<Q", x) for x in data)
    k = b"".join(struct.pack("<Q", x) for x in kunci)
    t = bytes(x ^ y for x, y in zip(a, k))
    return t.decode("utf-16le").split("\0")[0]

K = [0x7f69a30995266e95, 0xe45d52788e2a547a,
     0xddb11813b18b5557, 0xbbd2156a9962f664]
print(buka([0x7f01a36a95506ee6, 0xe473520c8e595415,
            0xddb11876b1f35532, K[3]], K))
print(buka([0x7f0ca36195456ec6, 0xe43852148e5f541e,
            K[2], K[3]], K))
```

```
svchost.exe
Schedule
```

9.10 Langkah 10, Menemukan Alamat Server

Langkah terakhir membuka konfigurasi di ujung file .bin dengan kunci 01 02 03 04 05 yang ditemukan di fungsi 0x8C6C.

08_konfigurasi.py

```
import sys, struct, re
from rc4x import rc4x

b = open(sys.argv[1], "rb").read()
stub = struct.unpack("<I", b[0:4])[0]
muat = struct.unpack("<I", b[4 + stub:8 + stub])[0]
pos = 8 + stub + muat
n = struct.unpack("<I", b[pos:pos + 4])[0]
konf = rc4x(bytes([1, 2, 3, 4, 5]), b[pos + 4:pos + 4 + n])
ip = re.findall(rb"(?:\d{1,3}\.){3}\d{1,3}", konf)
port = struct.unpack("<H", konf[0x100:0x102])[0]
print("Server :", ip[0].decode(), "port", port)
for m in re.finditer(rb"(?:[\x20-\x7e]\x00){4,}", konf):
    print(hex(m.start()), m.group().decode("utf-16le"))
```

```
Server : 43.156.35.124 port 1234
0x430 1116
0x670 Microsoft Compatibility system
0x864 C:\Program Files\Common Files
0x92c 444.exe
0x990 SbieDll.dll
0x9f4 SbieDll.bin
```

Pemilik alamat dapat dilihat dengan perintah *whois*. Perintah ini bertanya kepada lembaga pencatat alamat internet dan tidak menghubungi server pelaku.

```
$ whois 43.156.35.124 | grep -E "netname|country|route|origin"
netname:   ACEVILLEPTELTD-SG
country:   SG
route:     43.156.0.0/18
origin:    AS132203
```

9.11 Langkah 11, Memeriksa Area Tanda Tangan

Pada sampel Oktober, tanda bahaya pertama muncul saat daftar bagian DLL dibaca dengan pefile. Ada bagian bernama acak .yls3k4v, dan titik masuk DLL berada di ruang yang seharusnya kosong. Kode tambahan itu ternyata mencari muatan di area tanda tangan program induk. Skrip berikut mengambil isi area tersebut, mencari penanda CA FE BA BE, lalu menyimpan data setelahnya.

09_isi_tanda_tangan.py

```
import sys, pefile

b = open(sys.argv[1], "rb").read()
pe = pefile.PE(data=b)
sd = pe.OPTIONAL_HEADER.DATA_DIRECTORY[4]
blok = b[sd.VirtualAddress:sd.VirtualAddress + sd.Size]
print("area tanda tangan:", hex(sd.VirtualAddress),
      sd.Size, "byte")
k = blok.find(bytes.fromhex("cafebabe"))
print("penanda CA FE BA BE di posisi:", k)
open(sys.argv[2], "wb").write(blok[k + 8:])
```

```
area tanda tangan: 0xc8a00 41032 byte
penanda CA FE BA BE di posisi: 10752
```

Program dengan tanda tangan sah biasanya memiliki area tanda tangan yang hanya berisi sertifikat. Data tambahan dengan penanda khusus di dalamnya adalah tanda kuat penyalahgunaan.

9.12 Langkah 12, Membuka Tahap Dua

Data dari area tanda tangan didekripsi dengan RC4 standar memakai kunci *AtkExEmbedKey2026* yang ditemukan di bagian .yls3k4v. Fungsi RC4 standar hanya berbeda satu baris dari versi sampel November.

rc4.py

```
def rc4(kunci, data):
    """RC4 standar."""
    S = list(range(256))
    j = 0
```

```
    for i in range(256):
        j = (j + S[i] + kunci[i % len(kunci)]) & 255
        S[i], S[j] = S[j], S[i]
    i = j = 0
    hasil = bytearray()
    for c in data:
        i = (i + 1) & 255
        j = (j + S[i]) & 255
        S[i], S[j] = S[j], S[i]
        hasil.append(c ^ S[(S[i] + S[j]) & 255])
    return bytes(hasil)
```

10_tahap_dua.py

```
import sys
from rc4 import rc4

data = open(sys.argv[1], "rb").read()
buka = rc4(b"AtkExEmbedKey2026", data)
open(sys.argv[2], "wb").write(buka)
k = buka.find(b"SCURLTPL")
blok = buka[k + 8:k + 300]
url = bytes(c ^ 0x5A for c in blok)
url = url.split(b"\0")[0].lstrip(b"Z[")
print("ukuran tahap dua:", len(buka))
print("alamat unduhan:", url.decode())
```

```
ukuran tahap dua: 30272
alamat unduhan: https://blasidha.eu.cc/files/quoweu.eu.cc.bin
```

Alamat ini cukup dicatat dan dilaporkan. Jangan membuka alamat tersebut di browser dan jangan mengunduh isinya.

Bab 10

Memakai dan Melaporkan Temuan

Hasil investigasi baru bermanfaat bila dipakai untuk melindungi orang lain. Bab ini membahas cara mengubah temuan menjadi tindakan.

10.1 Indikator Kompromi

Indikator kompromi (*indicators of compromise*, IOC) adalah jejak yang bisa dipakai untuk mendeteksi serangan yang sama di tempat lain. Dari sampel ini, IOC yang paling berguna sebagai berikut.

Tabel 10.1 Indikator kompromi utama

Jenis	Nilai
Alamat server	43.156.35.124 port 1234/TCP
Domain unduhan	blasidha.eu.cc, khususnya /files/quoweu.eu.cc.bin
Nama file	SbieDll.dll dan SbieDll.bin di luar folder Sandboxie, 444.exe, CrashHunter_PC3.dll di luar folder program NetEase
Folder	C:\Program Files\Common Files\444.exe
Nama layanan atau tugas	Microsoft Compatibility system
Hash SHA-256	Lihat Lampiran B

10.2 Memakai IOC di Kantor

1. Blokir alamat 43.156.35.124 dan domain blasidha.eu.cc di *firewall* dan DNS kantor.
2. Cari koneksi ke alamat dan domain tersebut di catatan *firewall*, DNS, atau *router* untuk mengetahui apakah ada komputer yang sudah terinfeksi.
3. Cari file *SbieDll.dll*, *444.exe*, dan *CrashHunter_PC3.dll* di folder pengguna pada semua komputer Windows kantor.
4. Periksa daftar layanan dan tugas terjadwal yang bernama *Microsoft Compatibility system*.
5. Masukkan nilai *hash* ke daftar blokir antivirus atau perangkat EDR.

10.3 Contoh Aturan YARA

YARA adalah bahasa aturan untuk mengenali file berdasarkan pola. Aturan berikut mengenali file .bin dengan struktur seperti sampel ini, yaitu panjang stub 0xF98 diikuti pembuka fungsi x64.

```
rule Umpan_Faktur_SbieDll_Bin
{
  strings:
    $awal = { 98 0F 00 00 48 89 4C 24 08 48 83 EC 28 }
  condition:
    $awal at 0 and filesize < 2MB
}
```

Aturan kedua mengenali DLL yang ditambal seperti pada sampel Oktober, yaitu DLL yang memiliki bagian bernama .yls3k4v atau menyusun kunci AtkExEmbedKey2026 di memori.

```
import "pe"
rule DLL_Tambalan_Kunci_Embed
{
  strings:
    $k1 = { C7 44 24 20 41 74 6B 45 }
    $k2 = { C7 44 24 24 78 45 6D 62 }
  condition:
    pe.is_dll() and
```

```
    (pe.section_index(".yls3k4v") ⩾ 0 or ($k1 and $k2))
}
```

Aturan seperti ini perlu disesuaikan bila pelaku mengubah panjang stub pada kampanye berikutnya. Gunakan sebagai titik awal, bukan satu-satunya alat deteksi.

10.4 Melaporkan Server Pelaku

Penyedia layanan awan umumnya menindak server yang dipakai untuk kejahatan bila menerima laporan yang lengkap. Laporan ke alamat abuse@tencent.com sebaiknya memuat hal berikut.

- Alamat IP dan port, yaitu 43.156.35.124 port 1234/TCP.
- Penjelasan singkat bahwa alamat tersebut dipakai sebagai server pengendali program jahat.
- Nilai SHA-256 sampel dan cara konfigurasi ditemukan.
- Tanggal sampel diterima.

Domain seperti blasidha.eu.cc dapat dilaporkan ke URLhaus milik abuse.ch beserta alamat lengkapnya, agar penyedia keamanan dan peneliti lain ikut memblokirnya.

Jangan melampirkan file program jahat dalam email laporan. Cukup nilai *hash* dan penjelasan. Laporan juga bisa disampaikan ke komunitas berbagi ancaman seperti URLhaus atau ThreatFox dari abuse.ch agar peneliti lain dan penyedia keamanan ikut mengetahuinya.

10.5 Kapan Perlu Analisis Dinamis

Analisis statis di buku ini sudah menjawab pertanyaan utama. Analisis dinamis baru diperlukan bila ingin mengetahui perintah apa saja yang dikirim server atau program tambahan apa yang diunduh. Bila diperlukan, analisis dinamis hanya boleh dilakukan di mesin virtual yang terisolasi penuh dari jaringan kantor dan rumah, dengan *snapshot* yang bisa dikembalikan, dan sebaiknya

oleh tim yang berpengalaman. Menjalankan sampel dengan koneksi internet aktif akan membuat sampel menghubungi pelaku.

Bagian IV

MENCEGAH DAN MENANGANI

Pencegahan untuk pribadi dan kantor, serta langkah darurat bila terlanjur mengklik.

> **“There are only two types of companies: those that have been hacked and those that will be.”**
>
> *Hanya ada dua jenis perusahaan, yang sudah diretas dan yang akan diretas.*
>
> Robert S. Mueller III, Direktur FBI, RSA Conference, 1 Maret 2012



Bab 11

Langkah Pencegahan

Penyamaran berlapis pada Bagian II terdengar canggih. Namun seluruh rantai pada kedua sampel gagal bila satu langkah saja tidak terjadi, yaitu file program di dalam arsip tidak diklik. Karena modelnya sama, pencegahannya pun berlaku untuk varian apa pun. Pencegahan bisa dilakukan oleh setiap orang, dengan pengaturan sederhana di WhatsApp dan Windows, serta prosedur kerja di kantor.

11.1 Kebiasaan Pribadi

1. Jangan membuka lampiran dari nomor yang tidak dikenal, apalagi yang meminta dibuka di komputer.
2. Curigai setiap ZIP, RAR, 7Z, IMG, VHD, VBS, EXE, SCR, BAT, MSI, dan APK yang dikirim lewat chat. Dokumen kerja yang wajar dikirim dalam bentuk PDF, Word, atau Excel.
3. Abaikan nama file dan nomor pengirimnya. Yang perlu dikenali adalah polanya, yaitu tema keuangan, permintaan meneruskan ke atasan, dan permintaan membuka di komputer.
4. Bila pesan mengaku dari pemasok atau rekan kerja, hubungi orang itu lewat nomor yang sudah Anda simpan sebelumnya, atau lewat email resmi kantor. Jangan membalas ke nomor yang mengirim lampiran.
5. Jangan meneruskan file mencurigakan kepada atasan, sekalipun diminta. Laporkan dulu kepada tim TI atau atasan langsung tanpa melampirkan filenya.

6. Bila sebuah dokumen tidak terbuka setelah diklik, berhenti. Jangan mencoba lagi.

11.2 Pengaturan WhatsApp

Beberapa pengaturan WhatsApp membantu mengurangi pesan dari orang asing. Nama menu bisa sedikit berbeda tergantung versi aplikasi.

Tabel 11.1 Pengaturan WhatsApp yang disarankan

Pengaturan	Fungsi
Privasi, Grup, pilih Kontak Saya	Orang asing tidak bisa langsung memasukkan Anda ke grup penipuan.
Privasi, Lanjutan, blokir pesan dari akun tidak dikenal	Membatasi pesan dari akun asing yang mengirim pesan dalam jumlah banyak.
Privasi, Panggilan, bisukan penelepon tidak dikenal	Panggilan dari nomor asing tidak berdering.
Penyimpanan dan Data, matikan unduh otomatis untuk dokumen	Lampiran tidak tersimpan otomatis sebelum Anda memutuskan.
Perangkat Tertaut, periksa secara berkala	Memastikan tidak ada komputer asing yang terhubung ke WhatsApp Anda.
Verifikasi Dua Langkah, aktifkan PIN	Akun lebih sulit diambil alih.

Bila menerima pesan seperti contoh di buku ini, gunakan tombol **Laporkan** lalu **Blokir** di WhatsApp. Laporan membantu WhatsApp menutup nomor yang dipakai menyebarkan file berbahaya.

11.3 Pengaturan Windows

1. **Tampilkan ekstensi file.** Buka File Explorer, menu *View*, lalu centang *File name extensions*. Setelah itu file program akan terlihat berakhiran .exe dan tidak bisa menyamar sebagai faktur.

2. **Aktifkan Microsoft Defender dan SmartScreen.** Pastikan keduanya menyala dan diperbarui. Jangan mengabaikan peringatan SmartScreen saat membuka file dari internet.
3. **Gunakan akun pengguna biasa.** Untuk pekerjaan sehari-hari, jangan memakai akun *administrator*. Program jahat yang berjalan di akun biasa memiliki ruang gerak lebih sempit, terutama untuk teknik pencurian token seperti yang dimiliki sampel ini.
4. **Rutin memperbarui Windows dan browser.** Pembaruan menutup celah yang sering dimanfaatkan program jahat.
5. **Jangan menyimpan kata sandi penting di browser** pada komputer kerja bersama. Gunakan pengelola kata sandi dan aktifkan verifikasi dua langkah di setiap akun.

11.4 Prosedur Kantor dan Bagian Keuangan

Bagi perusahaan, pencegahan perlu dituangkan dalam prosedur tertulis agar tidak bergantung pada kewaspadaan satu orang.

Tabel 11.2 Prosedur yang disarankan untuk kantor

Area	Prosedur
Penerimaan tagihan	Tagihan hanya diterima lewat email resmi atau portal pemasok, tidak lewat WhatsApp pribadi staf.
Perubahan rekening	Setiap permintaan perubahan rekening pemasok wajib dikonfirmasi lewat telepon ke nomor yang tercatat di kontrak, dan disetujui dua orang.
Lampiran	Server email kantor memblokir lampiran program dan ZIP berisi program. Staf dilarang membuka file program dari chat.
Komputer keuangan	Komputer yang dipakai untuk *internet banking* tidak dipakai untuk membuka lampiran dari luar.
Pelaporan	Ada satu nomor atau alamat email internal untuk melaporkan pesan mencurigakan tanpa rasa takut disalahkan.

Area	Prosedur
Pelatihan	Contoh pesan seperti di buku ini dibagikan kepada staf secara berkala.

11.5 Perlindungan Teknis untuk Tim TI

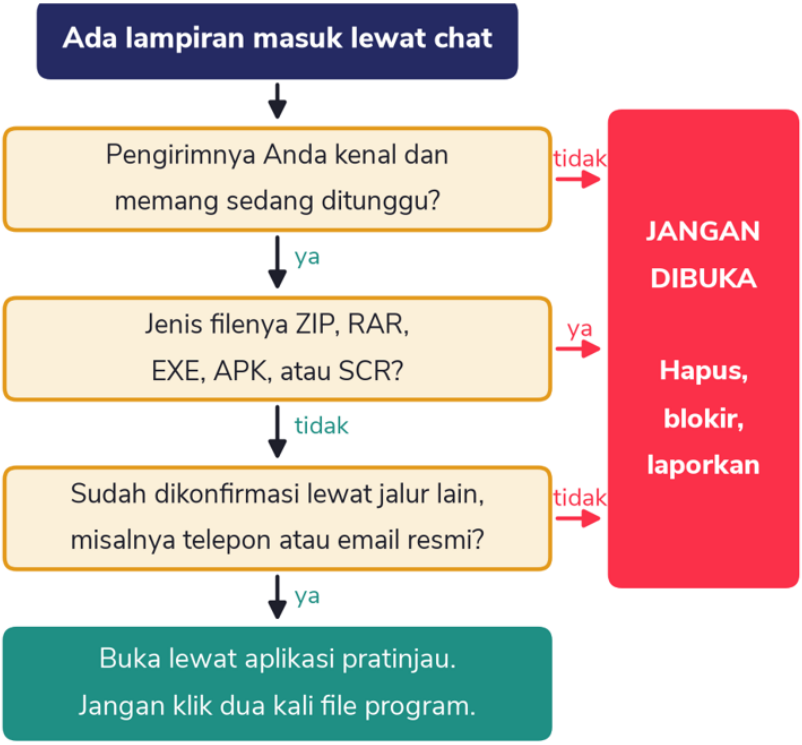
Tim TI dapat menutup celah teknis yang dimanfaatkan sampel ini secara langsung.

Tabel 11.3 Kontrol teknis terhadap tiap lapis serangan

Lapis yang dihadang	Kontrol
Program dari folder pengguna	Kebijakan *AppLocker* atau *Windows Defender Application Control* yang melarang program berjalan dari Downloads, Desktop, dan folder sementara.
DLL side-loading	Aturan deteksi untuk program bertanda tangan yang memuat DLL tidak bertanda tangan dari folder yang sama, terutama SbieDll.dll di luar folder Sandboxie.
Muatan di memori	Perangkat EDR yang memantau alokasi memori yang bisa dieksekusi dan pembuatan *thread* dari memori tersebut.
Muatan di tanda tangan	Pemeriksaan program bertanda tangan yang area tanda tangannya jauh lebih besar dari ukuran sertifikat biasa atau berisi data tambahan.
Bertahan di sistem	Pemantauan pembuatan layanan dan tugas terjadwal baru, khususnya yang memakai nama mirip Microsoft, serta pemasangan alat *remote monitoring and management* yang tidak dikenal.
File disk image	Kebijakan yang melarang pemasangan otomatis file .img dan .vhd dari unduhan, serta memblokir skrip .vbs.
Komunikasi keluar	*Firewall* yang membatasi koneksi keluar ke port tidak lazim seperti 1234.

11.6 Pohon Keputusan Sebelum Membuka Lampiran

Gunakan alur berikut setiap kali menerima lampiran lewat chat. Bila salah satu jawaban mengarah ke kotak merah, jangan buka file tersebut.



Gambar 11.1 Pohon keputusan sebelum membuka lampiran.



Bab 12

Bila Terlanjur Membuka

Kesalahan bisa terjadi pada siapa pun. Yang penting adalah bertindak cepat. Semakin cepat komputer diputus dari internet, semakin sedikit waktu yang dimiliki pelaku.



Gambar 12.1 Urutan langkah darurat setelah file terlanjur dibuka.

12.1 Menit Pertama

1. Putuskan komputer dari internet. Cabut kabel LAN dan matikan Wi-Fi.
2. Jangan mematikan komputer bila kantor memiliki tim TI. Bukti di memori komputer bisa membantu pemeriksaan. Bila tidak ada tim TI, komputer boleh dimatikan.
3. Jangan login ke *internet banking*, email, atau akun apa pun dari komputer tersebut.
4. Beri tahu tim TI atau atasan sekarang juga. Sebutkan nama file, jam membuka, dan siapa saja yang sudah menerima file itu.

12.2 Jam Pertama

Lakukan langkah berikut dari perangkat lain yang bersih, misalnya ponsel atau komputer lain.

1. Ganti kata sandi email, *internet banking*, dan akun penting lain. Mulailah dari email, karena email dipakai untuk memulihkan akun lainnya.
2. Di setiap akun, pilih keluar dari semua perangkat atau semua sesi agar *cookie* yang dicuri tidak bisa dipakai lagi.
3. Aktifkan verifikasi dua langkah bila belum.
4. Hubungi *call center* bank bila komputer itu pernah dipakai untuk transaksi, lalu minta pemantauan atau pemblokiran sementara.

12.3 Hari Pertama

1. Di WhatsApp, buka **Perangkat Tertaut** dan keluarkan semua perangkat.
2. Kabari rekan kerja dan pemasok bahwa akun Anda mungkin disalahgunakan, dan minta mereka mengabaikan lampiran yang seolah-olah dikirim dari Anda.
3. Periksa riwayat transaksi rekening dan kartu kredit.

4. Bila Anda sudah meneruskan file itu, beri tahu penerimanya agar tidak membukanya.

12.4 Pemeriksaan oleh Tim TI

Tim TI dapat memeriksa jejak spesifik dari sampel ini pada komputer yang dicurigai.

Tabel 12.1 Jejak yang perlu diperiksa

Lokasi	Yang dicari
C:\Program Files\Common Files	File 444.exe, SbieDll.dll, dan SbieDll.bin
Layanan dan Task Scheduler	Entri bernama Microsoft Compatibility system
Daftar koneksi jaringan	Koneksi ke 43.156.35.124 port 1234, misalnya dengan perintah *netstat -ano*
Catatan DNS	Permintaan ke domain blasidha.eu.cc atau subdomain eu.cc lain yang tidak dikenal
Folder hasil ekstraksi	File CrashHunter_PC3.dll, 2026Pmksnflacker.exe, atau program bertanda tangan lain yang berdampingan dengan DLL
Folder Downloads dan Documents	File bernama faktur, ringkasan, saldo, atau tagihan yang ternyata berakhiran .exe
Catatan firewall atau router kantor	Riwayat koneksi ke 43.156.35.124 sejak sampel diterima

12.5 Sesudahnya

Komputer yang sudah menjalankan program pengendali jarak jauh tidak bisa dipercaya sepenuhnya walaupun antivirus menyatakan bersih, karena pelaku mungkin sudah memasang program tambahan. Cara paling aman adalah mencadangkan dokumen penting, memasang ulang Windows dari awal, lalu memeriksa dokumen cadangan dengan antivirus sebelum dipakai lagi.

12.6 Ke Mana Melapor

Tabel 12.2 Saluran pelaporan

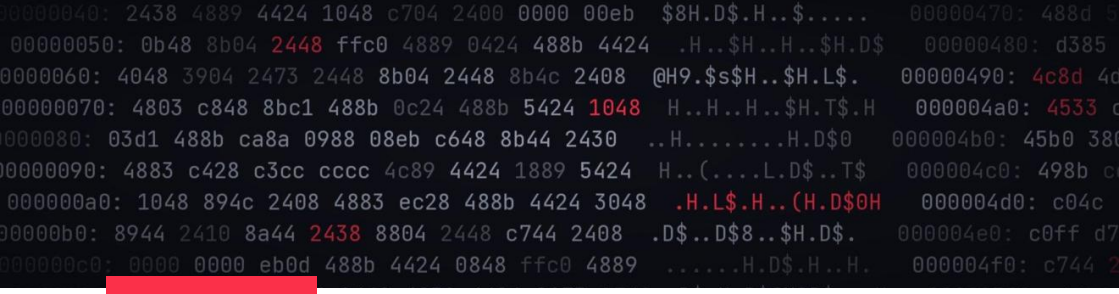
Saluran	Untuk apa
Fitur Laporkan di WhatsApp	Melaporkan nomor pengirim agar ditinjau dan ditutup oleh WhatsApp.
aduannomor.id	Layanan Komdigi untuk melaporkan nomor telepon yang dipakai menipu.
cekrekening.id	Melaporkan dan memeriksa rekening bank yang terkait penipuan.
Bank dan OJK	Bila uang sudah berpindah, segera hubungi bank. Korban penipuan keuangan juga dapat melapor ke OJK melalui Indonesia Anti-Scam Centre (IASC).
patrolisiber.id	Laporan kejahatan siber ke Direktorat Tindak Pidana Siber Bareskrim Polri. Laporan resmi tetap bisa dibuat di kantor polisi terdekat.
abuse@tencent.com	Laporan penyalahgunaan server 43.156.35.124 kepada penyedia layanan awannya.

Simpan bukti sebelum menghapus apa pun. Tangkapan layar pesan, nomor pengirim, nama file, jam kejadian, dan bukti transaksi akan dibutuhkan saat melapor.

Lampiran

LAMPIRAN

Daftar periksa, indikator teknis, glosarium, dan daftar pustaka.



Lampiran A

Daftar Periksa Lampiran Chat

Halaman ini bisa dicetak dan ditempel di meja kerja.

Tabel A.1 Daftar periksa sebelum membuka lampiran

☐	Apakah saya mengenal pengirimnya dan nomornya sudah tersimpan sebelumnya?
☐	Apakah saya memang sedang menunggu file ini?
☐	Apakah filenya PDF, Word, atau Excel, dan tidak berupa ZIP, RAR, EXE, APK, atau SCR?
☐	Apakah pesannya tidak memaksa, tidak terburu-buru, dan tidak meminta dibuka di komputer?
☐	Apakah saya sudah mengonfirmasi lewat jalur lain, misalnya telepon ke nomor yang tercatat?
☐	Apakah ekstensi file di komputer saya sudah ditampilkan?

Bila ada satu saja jawaban **tidak**, jangan buka file tersebut. Hapus, blokir pengirimnya, dan laporkan.

Tabel A.2 Bila terlanjur membuka

No	Langkah
1	Cabut kabel LAN dan matikan Wi-Fi.
2	Beri tahu tim TI atau atasan.
3	Dari ponsel atau komputer lain, ganti sandi email dan *internet banking*, lalu keluar dari semua sesi.
4	Keluarkan semua perangkat di menu Perangkat Tertaut WhatsApp.
5	Hubungi bank bila komputer pernah dipakai bertransaksi.
6	Pasang ulang Windows dan laporkan kejadiannya.

Lampiran B

Indikator Teknis Lengkap

Nilai *hash* berikut adalah SHA-256 dari file kedua sampel yang diperiksa. Nilai untuk program inti dihitung dari hasil rekonstruksi dengan skrip pada Langkah 8, sehingga dapat berbeda bila metode rekonstruksinya berbeda.

Tabel B.1 Sidik digital sampel November (SHA-256)

File	SHA-256
Balance_by_Invoice_20261103.zip	388febbcf46a9e682fdaa2a8821504822db82a3f96de8ac3a3de387eb8b6c24a
Ringkasan_Faktur_Pemasok_20261116.exe	185b7a487316454da04e9cc0fe6eb370bb2955cf6096fe3e8c02c46f8989ba37
SbieDll.dll	c73986636f021d21d55a855aa5d138d1e870945e04a0cc494950eee759351c96
SbieDll.bin	97a048f30f033b183faa88668f9800390bc616372fe0b418f68648e89040eb09
Program inti (rekonstruksi)	a6b1a9593bbb5514578cdc1b3c45d057a8b1d78b25b1e021e4d5f2b7378f3e1c

Tabel B.2 Sidik digital sampel Oktober (SHA-256)

File	SHA-256
Ringkasan_20261019_Saldo_Tagihan.zip	57eb175eab739590e173db95f2a25157549ed2e3261832e0b04798cfc8e33f30
2026Pmksnflacker.exe	b22c93ce3694b8ec6a4aab5933e9642cd4de86f5ead6cf6c37afc628ff860f64

File	SHA-256
CrashHunter_PC3.dll	`7e1fdf98de48b6a89de8d5c5fbf92c4bc869aeb3b65b958a2b2f085bd5e7c6d9`
Tahap dua (hasil dekripsi)	`b1759ac8292eabbbdd165bf37866062c20c97fd2527f740d09dcbd39435d74e2`

Tabel B.3 Indikator jaringan dan sistem

Jenis	Nilai
Server pengendali	43.156.35.124:1234/TCP (sampel November)
URL unduhan	https://blasidha.eu.cc/files/quoweu.eu.cc.bin (sampel Oktober)
Jaringan	AS132203, Aceville Pte. Ltd. (Tencent Cloud), Singapura
Protokol	WebSocket (libwebsockets 3.2.3), data dikompresi zlib 1.2.13
Tanda kampanye	1116
Nama penyamaran	Microsoft Compatibility system
Jalur pemasangan	C:\Program Files\Common Files\444.exe
File pendamping	SbieDll.dll, SbieDll.bin

Tabel B.4 Parameter teknis untuk penyelidik

Parameter	Nilai
Panjang stub	0xF98 (3.992 byte), diawali 48 89 4C 24 08
Algoritma muatan	RC4 modifikasi, byte kunci aliran S[i] XOR S[j]
Kunci muatan	210 byte, tersimpan di file, sekaligus kepala PE (e_lfanew 0x138)
Kunci konfigurasi	01 02 03 04 05, RC4 modifikasi yang sama
Ukuran konfigurasi	0xA88 (2.696 byte)
Fungsi penting	Stub 0x240 (utama), 0xF3C (dekripsi). Program inti 0x6860 (DllMain), 0x6390 (utama), 0xA4E8 (muat konfigurasi), 0x8C6C (dekripsi konfigurasi)

Parameter	Nilai
Waktu kompilasi	Umpan 1 Oktober 2024, pemuat 25 Juli 2025, program inti 6 Maret 2026
Sampel Oktober, bagian sisipan	.yls3k4v (0x342000), dipanggil dari titik masuk baru 0x228D98
Sampel Oktober, kunci	AtkExEmbedKey2026, RC4 standar
Sampel Oktober, penanda	CA FE BA BE di area tanda tangan, posisi 10.752
Sampel Oktober, URL	Penanda SCURLTPL, XOR 0x5A

Glosarium

Analisis dinamis
Pemeriksaan program jahat dengan menjalankannya di lingkungan terkendali.

Analisis statis
Pemeriksaan file dengan membaca isinya tanpa menjalankannya.

Authenticode
Sistem tanda tangan digital untuk program Windows. Area tempat tanda tangan disimpan tidak ikut dihitung saat keaslian program diperiksa.

Backdoor
Program yang membuka jalan masuk tersembunyi bagi pelaku untuk mengendalikan komputer korban.

C2
Command and control, server yang dipakai pelaku untuk mengirim perintah dan menerima data dari komputer korban.

Cookie sesi
Data kecil di browser yang menandakan Anda sudah login. Bila dicuri, orang lain bisa masuk ke akun tanpa kata sandi.

DLL
Dynamic-link library, file pustaka yang dimuat oleh program Windows saat berjalan.

DLL side-loading
Teknik menaruh DLL palsu di samping program asli agar program asli memuat dan menjalankan kode jahat.

EDR

Endpoint detection and response, perangkat lunak keamanan yang memantau perilaku program di komputer.

Ekstensi file

Akhiran nama file yang menunjukkan jenisnya, misalnya .pdf, .xlsx, atau .exe.

Entropi

Ukuran keacakan isi file dalam bit per byte. Nilai yang mendekati 8 menandakan data terenkripsi atau terkompresi.

Hash

Sidik digital file. File yang sama selalu menghasilkan *hash* yang sama.

Hash nama API

Teknik mencari fungsi Windows dengan membandingkan nilai *hash* namanya, sehingga nama fungsi tidak pernah tertulis sebagai teks.

IOC

Indicator of compromise, jejak yang menandakan sebuah sistem telah diserang.

Little-endian

Cara menyimpan angka dengan byte terkecil di depan, dipakai oleh prosesor Intel dan AMD.

Muatan

Bagian utama program jahat yang dibawa dan dijalankan oleh pemuat.

PE

Portable Executable, format file program dan DLL di Windows.

PEB

Process Environment Block, struktur data Windows yang berisi informasi tentang program yang sedang berjalan.

RAT

Remote access trojan, program jahat yang memberi pelaku kendali jarak jauh atas komputer korban.

RC4

Algoritma enkripsi aliran yang sederhana dan cepat, sering dipakai program jahat untuk menyembunyikan data.

Reflective loading

Teknik memuat program langsung dari memori tanpa menyimpannya sebagai file.

Shellcode

Potongan kode mesin mandiri yang bisa berjalan di mana saja di memori.

Stack string

Teknik menyusun teks sedikit demi sedikit di memori saat program berjalan agar teks tidak terlihat dalam file.

Tanda tangan digital

Penanda bahwa sebuah program dibuat oleh pengembang tertentu dan belum diubah sejak ditandatangani.

WebSocket

Protokol komunikasi dua arah yang terus terbuka antara komputer dan server.

XOR

Operasi logika yang sering dipakai untuk menyamarkan data. Data yang di-XOR dua kali dengan kunci yang sama akan kembali seperti semula.

YARA

Bahasa aturan untuk mengenali file berdasarkan pola isi.

Daftar Pustaka

abuse.ch. (2026). *ThreatFox and URLhaus*. https://abuse.ch

CXOToday. (2026, 15 September). *Quick Heal warns of evolving WhatsApp malware targeting finance teams and executives*. https://cxotoday.com/cybersecurity/quick-heal-warns-of-evolving-whatsapp-malware-targeting-finance-teams-and-executives/

Microsoft. (2023). *Windows Authenticode portable executable signature format*. Microsoft Learn. https://learn.microsoft.com/windows-hardware/drivers/install/authenticode

Microsoft. (2024). *Dynamic-link library search order*. Microsoft Learn. https://learn.microsoft.com/windows/win32/dlls/dynamic-link-library-search-order

MITRE. (2024). *Hijack execution flow: DLL side-loading (T1574.002)*. MITRE ATT&CK. https://attack.mitre.org/techniques/T1574/002/

MITRE. (2024). *Reflective code loading (T1620)*. MITRE ATT&CK. https://attack.mitre.org/techniques/T1620/

Sikorski, M., & Honig, A. (2012). *Practical malware analysis: The hands-on guide to dissecting malicious software*. No Starch Press.

VirusTotal. (2026). *YARA documentation*. https://virustotal.github.io/yara/

WhatsApp. (2026). *Tentang keamanan dan privasi di WhatsApp*. Pusat Bantuan WhatsApp. https://faq.whatsapp.com

Zscaler ThreatLabz. (2024). *DodgeBox: A deep dive into the updated arsenal of APT41, part 1*. https://www.zscaler.com/blogs/security-research/dodgebox-deep-dive-updated-arsenal-apt41-part-1

Tentang Penulis



Liu Purnomo, S.Kom. adalah praktisi teknologi dan *software engineer* yang telah berkecimpung di dunia pesawat tanpa awak sejak 2013 dan menjadi instruktur drone profesional sejak 2014. Ia memegang sertifikasi Master Instructor dari Badan Nasional Sertifikasi Profesi (BNSP) serta sertifikasi Remote Pilot dari Direktorat Jenderal Perhubungan Udara.

Selain melatih pilot dan mengembangkan aplikasi untuk komunitas drone, ia menulis sejumlah buku, di antaranya *The Remote Pilot* dan *Seni dan Filosofi Penerbangan Drone*. Ebook ini ia tulis setelah menerima sendiri pesan berisi lampiran faktur palsu yang dibahas di dalamnya, lalu membedahnya agar orang lain tidak menjadi korban.

Pembaca yang menerima pesan serupa, menemukan varian baru dari modus ini, atau ingin menyampaikan masukan dapat menghubunginya melalui saluran berikut.

Situs liupurnomo.com
Email hi@liupurnomo.com

liupurnomo

SEBARKAN EBOOK INI.

Satu orang yang tahu bisa mencegah satu kantor menjadi korban. Bagikan ebook ini kepada rekan kerja, keluarga, dan grup WhatsApp Anda.

liupurnomo.com
hi@liupurnomo.com