A Review of NTLM Rainbow Table Generation Techniques

AN Manager atau LM adalah protokol otentikasi yang dirancang untuk memaksimalkan keamanan sandi dalam lingkungan berbasis Windows. Protokol LM pertama kali digunakan dalam Produk Manajer LAN Microsoft sejak lama dan masih menjadi protokol otentikasi pilihan untuk sistem operasi yang lebih lama, seperti Windows 95 dan Windows NT 3.51 dan yang lebih lama. Kemudian, ketika Windows NT diperkenalkan, LM ditingkatkan dan diganti namanya menjadi protokol otentikasi NTLM.

a) Kelemahan Utama NTLM

- SAM memiliki beberapa kerentanan, yang memungkinkan penyerang mengakses sandi yang di-hash.
- NTLM dapat menggunakan maksimal 14 karakter untuk membuat hash yang disimpan dan dibagi menjadi dua string tujuh karakter.
- NTLM tidak dapat menggunakan huruf kecil.
- Algoritma hash yang digunakan untuk menyimpan kata sandi menjadi terkenal. Tindakan itu memungkinkan penyerang menebak sandi pengguna dengan menjalankan tebakan kata sandi melalui hash hingga hasilnya cocok dengan hasil yang disimpan di SAM.
- NTLM menggunakan mekanisme yang dikenal sebagai otentikasi pass-through untuk mendistribusikan tugas otentikasi. Cara melewati otentikasi dirancang menciptakan kemacetan di kontroler domain primer (PDC) dari setiap domain. Beberapa tugas yang dilakukan oleh PDC, seperti perubahan kata sandi, tidak dapat dipindahkan ke server lain.
- NTLM tidak memberikan cara bagi pengguna untuk memverifikasi bahwa server yang mereka sambungkan adalah yang ingin mereka sambungkan.
- NTLM sebagian besar terbatas pada interoperabilitas dengan produk Microsoft.
- NTLM tidak menyediakan cara bagi aplikasi tingkat menengah untuk mengakses sumber daya atas nama pengguna.

Rainbow Table adalah cara melakukan kriptanalisis dengan sangat cepat dan efisien. Jika seorang peretas telah memperoleh basis data nama pengguna dan kata sandi terenkripsi. Sistem mengkodekan kata sandi menggunakan fungsi hash, yang pada dasarnya merupakan cara untuk mengkondensasi kumpulan data tertentu menjadi string yang dipadatkan.

Saat mencari hash, komputasi tambahan diperlukan, tetapi komputasi yang diperlukan untuk pencarian secara signifikan lebih kecil dari jumlah yang diperlukan untuk praperhitungan. Seperti banyak algoritma, ada batasan dengan tabel pelangi. Tabel pelangi menghasilkan probabilitas keberhasilan yang sangat tinggi dan waktu pencarian lebih sedikit. Rantai yang lebih panjang membutuhkan lebih sedikit ruang penyimpanan, tetapi membutuhkan lebih banyak komputasi dan lebih banyak waktu untuk memecahkan kata sandi.

a) Pekerjaan Terkait

MARTIN E. HELLMAN, [4], dalam "A Cryptanalytic Time - Memory Trade-Off", menjelaskan bahwa metode probabilistik disajikan di mana crypt menganalisis setiap kriptosistem kunci N dalam operasi $N^{2/3}$dengan kata-kata $N^{2/3}$memori (nilai rata-rata) setelah prakomputasi yang membutuhkan operasi N. Metode ini Bekerja dalam serangan teks biasa yang dipilih dan, jika rantai blok cipher tidak digunakan, juga dapat digunakan dalam serangan hanya teks sandi. K bit pertama dari aliran kuncinya diambil sebagai fungsi f (K), di mana k adalah jumlah bit kunci. Ini dapat dilakukan di bawah serangan teks biasa yang dikenal.

Standar yang diusulkan mencakup ketentuan untuk rantai blok sandi dengan indikator acak. Meskipun teknik kriptanalitik trade-off timemory ini dapat dengan mudah digagalkan, teknik ini bekerja pada DES dalam mode blok dasar, yang lebih penting; ini menunjukkan bahwa ketika rantai blok cipher atau teknik lain ditambahkan, ukuran kunci yang lebih besar diperlukan untuk memiliki jaminan keamanan yang memadai.

Philippe Oechslin, [5], dalam “Making a Faster Cryptanalytic Time-Memory Trade Off”, menjelaskan bahwa Pada tahun 1980 Martin Hellman menggambarkan trade-off memori kriptanalitik, lalu ditingkatkan oleh Rivest sebelum tahun 1982 namun tidak terdapat pengoptimalan baru. Penulis mengusulkan cara baru untuk menghitung sebelumnya data yang mengurangi dua jumlah perhitungan yang diperlukan selama kriptanalisis sehingga mengurangi biaya overhead karena panjang rantai variabel, yang sekali lagi secara signifikan mengurangi jumlah perhitungan. Pengoptimalan kami memiliki sifat yang sama dengan penggunaan titik yang dibedakan, yaitu mengurangi jumlah pencarian tabel dengan faktor yang sama dengan panjang rantai. Untuk tingkat keberhasilan yang setara, metode kami mengurangi jumlah kalkulasi yang diperlukan untuk kriptanalisis dengan faktor dua dibandingkan metode asli dan dengan faktor yang bahkan lebih penting (12 dalam percobaan kami) terhadap poin yang dibedakan.

Panjang variabel rantai dibatasi oleh poin-poin penting yang menghasilkan lebih banyak alarm palsu dan lebih banyak *overhead*. Hal itu diduga dengan parameter yang berbeda keuntungannya bisa jauh lebih besar daripada faktor 12 ditemukan dalam percobaan. Fakta-fakta ini membentuk metode pengganti untuk meningkatkan metode aslinya dengan poin yang berbeda. Fakta bahwa metode tersebut menghasilkan rantai yang memiliki panjang konstan juga sangat menyederhanakan analisis metode dibandingkan dengan panjang variabel rantai menggunakan titik-titik yang berbeda. Panjang konstan bahkan bisa terbukti menguntungkan implementasi perangkat keras. Percobaan menunjukkan *trade-off time-memory* memungkinkan siapa pun yang memiliki komputer pribadi modern untuk memecahkan sistem kriptografi. Ini menunjukkan pentingnya menghapus sistem kriptografi lama secara bertahap jika lebih baik sistem ada untuk menggantikannya.

Hans Hedbom dkk, pada paper "*A Comparison of the Security of Windows NT and UNIX*", menjelaskan mengenai perbandingan dari dua operasi sistem, Windows NT dan UNIX. Penelitian ini membandingkan utama fitur keamanan dan kemudian membuat perbandingan pilihan kerentanan. Makalah ini menunjukkan bahwa keamanan mekanisme Windows NT sedikit lebih baik daripada dari UNIX. Terlepas dari kenyataan ini, kedua sistem menampilkan kerentanan serupa. Jorgen Blakstad dkk, pada paper "*All in a day's work: Password cracking for the rest of us*", menjelaskan bahwa mayoritas sistem komputer masih dilindungi terutama dengan Username dan password, dan banyak pengguna menggunakan kata sandi yang sama di beberapa sistem. Makalah ini menjelaskan eksperimen untuk menguji kekuatan dari pilihan kata sandi saat dikonversi ke hash LM, NT, dan MD5. Kesimpulannya adalah bahwa sejumlah besar kata sandi dapat di*crack* dalam keseharian, dan itu semua LM kata sandi hash dapat dipulihkan dengan mudah. Penggunaan fungsi hash yang lemah seperti itu dalam proses pengguna otentikasi dalam sistem operasi menimbulkan ancaman signifikan terhadap keamanan organisasi.

Manfaat utama dari *rainbow table* adalah ketika sebuah rainbow table memerlukan banyak waktu untuk memecahkan hash, setelah selesai dibuat tabel tersebut dapat digunakan berulang kali. Serangan dengan *rainbow table* lebih cepat dibandingkan dengan *brute force* dan membutuhkan alokasi memori yang lebih sedikit daripada serangan *full dictionary*. Dalam tulisan ini mengulas beberapa karya terpenting dalam pembuatan *rainbow table* dan menggunakannya pada windows NT, yaitu melawan NTLM serta membahas bagaimana NTLM lemah melawan serangan *rainbow table*.