Gapz and Redyms droppers based on Power Loader code

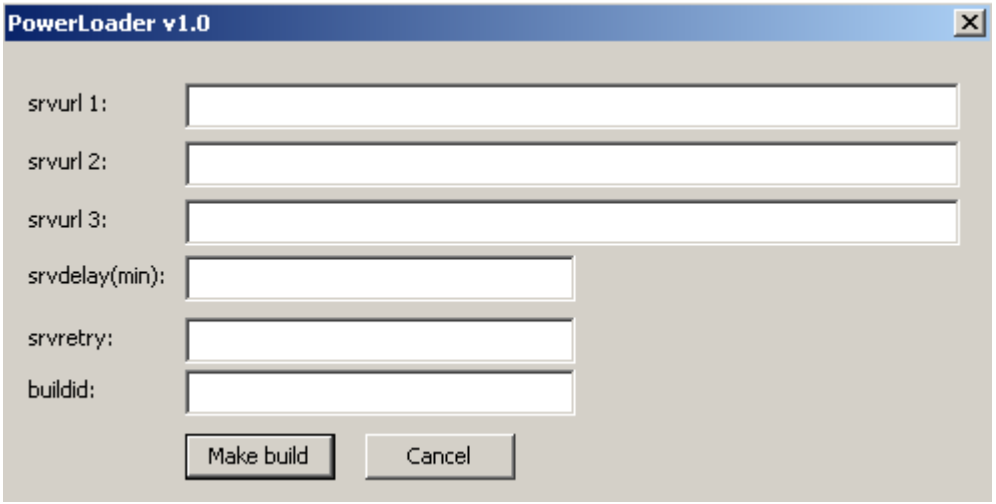
By Aleksandr Matrosov

Malware

Technical analysis of Power Loader, a special bot builder for making downloaders for other malware families and yet another example of specialization and modularity in malware production.

19 Mar 2013 • , 2 min. read

Power Loader is a special bot builder for making downloaders for other malware families and yet another example of specialization and modularity in malware production. The first time Power Loader was detected was in September 2012, using the family detection name Win32/Agent.UAW. This bot builder has been used for developing Win32/Gapz droppers (Win32/Gapz: steps of evolution) since October 2012. Starting from November 2012, the malware known as Win32/Redyms (What do Win32/Redyms and TDL4 have in common?) used Power Loader components in its own dropper. The price for Power Loader in the Russian cybercrime market is around $500 for one builder kit with C&C panel. (The image at the top of this post is the product logo used by the seller.)



The first version of the Power Loader builder was compiled at the beginning of September 2012. The time stamp of the compiled file is presented here:

Field Name	Data Value	Description
Machine	014Ch	i386®
Number of Sections	0004h	
Time Date Stamp	504EF332h	11/09/2012 08:15:46
Pointer to Symbol Table	00000000h	
Number of Symbols	00000000h	
Size of Optional Header	00E0h	
Characteristics	0102h	
Magic	010Bh	PE32
Linker Version	0009h	9.0

Power Loader uses one main C&C URL and two reserve URL's. All configuration data is stored into the .cfg section of the executable file. Configuration information is stored in plain text format, not encrypted.

```
}     `   [main]
srvurls=http://ripnhuipn.ru/power/c1.php;http://fikuskalus.ru/power/c1.php;
srvdelay=15
srvretry=2
buildid=test
```

The bot identifier is based on the unique MachineGuid value, which is stored in the system registry using random alphabetical symbols. This bot identifier is used to create the mutex and identify infection status.

```
char __stdcall GetMachineGuid(int MachineGuid)
{
  char v1; // bl@1
  int v3; // [sp+4h] [bp-Ch]@1
  int guid_size; // [sp+8h] [bp-8h]@1
  int v5; // [sp+Ch] [bp-4h]@1

  guid_size = 260;
  v3 = 1;
  v1 = 0;
  if ( !RegOpenKeyExA(0x80000002, "Software\\Microsoft\\Cryptography", 0, 0x101, &v5) )
  {
    if ( !RegQueryValueExA(v5, "MachineGuid", 0, &v3, MachineGuid, &guid_size) )
      v1 = 1;
    RegCloseKey(v5);
  }
  return v1;
}
```

Different dropper families have different export tables after unpacking the original dropper executable. The first version of the Power Loader export table looks like this:

Name	Address	Ordinal
DownloadRunExeId	00403E7B	1
DownloadRunExeUrl	00403D6C	2
DownloadUpdateMain	00403EC6	3
InjectApcRoutine	004036CF	4
InjectNormalRoutine	004036B4	5
SendLogs	00403F66	6
WriteConfigString	00403F39	7
start	00403CA7	

In the first version we didn't recognize the code injection method used to bypass HIPS used in Gapz. But the second version of Power Loader has special markers for the code injection method which is described at the beginning and the end of the shellcode. The export table is presented here:

Name	Address	Ordinal
DownloadRunExeId	004060D0	1
DownloadRunExeUrl	00405F80	2
DownloadUpdateMain	00406120	3
GetProcAddress64(void *,char *)	00403400	4
Inject32End	00404780	5
Inject32Normal	00404680	6
Inject32Start	00404710	7
InjectNormRoutine	004057A0	8
SendLogs	004061E0	9
WriteConfigString	004061B0	10
start	00405E30	

In the case of Win32/Redyms the export table looks like this:

Name	Address	Ordinal
CmdRunExeUrl	00405849	1
CmdSendLogs	004058D2	2
CmdUpdateMain	004058A4	3
CmdUpdateOption	00405892	4
InjectApcRoutine	00404607	5
InjectNormalRoutine	004045EC	6
InjectedShellCodeEnd	00404D8B	7
InjectedShellCodeStart	00404D3A	8
start	00404B5F	

This method of injecting code into explorer.exe is used for bypassing HIPS detection and is based on a technique for injecting code into a trusted process. More details have already been published one of my previous blog posts (Win32/Gapz: steps of evolution) and French researcher Axel Souchet published the PoC code for this technique.

One more interesting fact is that Power Loader uses the open source disassembler “Hacker Disassembler Engine” (also known as HDE) for code injection. And the same engine is used by Win32/Gapz in one of the bootkit shellcode modules. This doesn't prove that the developer of Power Loader and Gapz is the same person, but is nevertheless an interesting finding.
We continue our research and will be back soon with more interesting information.

Aleksandr Matrosov, Security Intelligence Team Lead

```
SHA1 hashes for analyzed samples:
Power Loader v1 (builder)     - a189ee99eff919b7bead989c6ca252b656b61137
Power Loader v1 (dropper)     - 86f4e140d21c97d5acf9c315ef7cc2d8f11c8c94
Power Loader v2 (dropper)     - 7f7017621c13065ebe687f46ea149cd8c582176d
```

Source: https://www.welivesecurity.com/2013/03/19/gapz-and-redyms-droppers-based-on-power-loader-code/