AtomBombing利用分析

嘶吼roartalk / 2018-01-16 15:23:45 / 浏览数 2309

0x00 前言

2016年10月，网络安全公司EnSilo的研究团队公开了一个支持所有Windows系统的代码注入方法，将其命名为AtomBombing。据说该方法能够绕过大多数的安全软件，并

于是，本文将要根据开源代码和资料，学习原理，测试功能，分析利用思路，总结防御方法

学习资料：

https://blog.ensilo.com/atombombing-brand-new-code-injection-for-windows

作者：Tal Liberman

POC：

https://github.com/BreakingMalwareResearch/atom-bombing/

0x01 简介

本文将要介绍以下内容：

- AtomBombing实现方法
- 关键技术
- 防御思路

0x02 基础知识

1、Atom Table

是一个存储字符串和相应标识符的系统定义表

应用程序将一个字符串放入一个Atom表中，并接收一个16位整数(WORD)作为标识(称为Atom)，可通过该标识访问字符串内容，实现进程间的数据交换

分类：

(1) Global Atom Table

所有应用程序可用

当一个进程将一个字符串保存到Global Atom
Table时，系统生成一个在系统范围内唯一的atom，来标示该字符串。在系统范围之内所有的进程都可以通过该atom(索引)来获得这个字符串,从而实现进程间的数据交换

(2) Local Atom Table

只有当前程序可用，相当于定义一个全局变量，如果程序多次使用该变量，使用Local Atom Table仅需要一次内存操作

参考资料：

https://msdn.microsoft.com/en-us/library/ms649053

常用API：

添加一个Global Atom：

```
ATOM WINAPI GlobalAddAtom(_In_ LPCTSTR lpString);
```

删除一个Global Atom：

```
ATOM WINAPI GlobalDeleteAtom(_In_ ATOM nAtom);
```

查找指定字符串对应的Global Atom：

```
ATOM WINAPI GlobalFindAtom(_In_ LPCTSTR lpString);
```

获取指定atom对应的字符串：

```
UINT WINAPI GlobalGetAtomName(
 _In_  ATOM   nAtom,
 _Out_ LPTSTR lpBuffer,
 _In_  int    nSize
);
```

注：

使用实例可参考如下连接：

https://github.com/sinmx/Windows2K/blob/661d000d50637ed6fab2329d30e31775046588a9/private/windows/base/client/tatom.c

2、APC注入

APC全称asynchronous procedure call，即异步过程调用

APC注入原理：

当线程处于警戒状态时，会检查APC队列，如果APC队列被插入函数指针，该函数将会得到执行

APC注入细节：

(1)
当线程调用SleepEx、SignalObjectAndWait、MsgWaitForMultipleObjectsEx，WaitForMultipleObjectsEx或者WaitForSingleObjectEx函数时，会切换到警戒状态(alertable state)

注：

警戒状态可参考：

https://msdn.microsoft.com/en-us/library/windows/desktop/aa363772(v=vs.85).aspx.aspx
"https://msdn.microsoft.com/en-us/library/windows/desktop/aa363772(v=vs.85).aspx")

(2) 当线程进入警戒状态时，会循环检查线程中的APC队列，如果APC队列中存在函数指针，那么就会调用该函数

(3) 使用QueueUserAPC函数向APC队列插入函数指针Loadlibrary()，实现加载DLL

(4) 注入成功后，警戒状态结束，程序继续运行，有可能造成程序不稳定，导致程序崩溃

(5) 如果没有删除APC队列，不能反复注入同一函数

(6) 使用APC注入，需要目标进程中至少有一个线程处于警戒状态或者能够进入警戒状态，否则无法实现APC注入

注：

大部分系统进程都满足条件，支持APC注入

可供参考的APC注入代码：

https://github.com/3gstudent/Inject-dll-by-APC

3、shellcode

在漏洞利用中，shellCode是指输入到存在漏洞的程序中的代码

相当于一个二进制代码框架，最终会将程序的流程跳转到payload

4、payload

主要功能代码(常见的如下载执行、反弹shell、新建用户等)，包含在shellCode中

0x03 实现方法

1、将任意数据写入目标进程地址空间中的任意位置(Write-What-Where)

通过读写atom向目标进程传递shellcode

自身进程通过GlobalAddAtom将shellcode添加到Global Atom Table中，目标进程调用GlobalGetAtomName即可从Global Atom Table中获取shellcode

所以接下来的关键是如何使目标进程调用GlobalGetAtomName

Tal Liberman的思路是通过APC注入，使目标进程调用GlobalGetAtomName

但是这里遇到了一个难题，QueueUserAPC函数只能向目标进程传入一个参数，而GlobalGetAtomName需要三个参数

于是Tal Liberman调试了QueueUserAPC函数，发现通过NtQueueApcThread函数能够传递三个参数

该问题得到解决

2、执行shellcode

目标进程调用GlobalGetAtomName从Global Atom Table中获取shellcode后，需要先保存shellcode再执行

第一种实现方法： 找到一段RWX的内存存储并执行

不通用，目前的系统保护机制很难找到这样的内存空间

第二种实现方法： 调用VirtualAllocEx分配一段内存

常用方法

注:

其他常见方法如通过VirtualProtect将shellcode的内存属性设置为可读可写可执行，然后跳到shellcode继续执行在这里的效果并不好，因为需要考虑使用QueueUserAPC函

所以Tal Liberman尝试了第三种方法： 找到一段RW的内存写入数据，构造ROP链实现shellcode的执行

寻找一段RW的内存并不难，Tal Liberman选择了KERNELBASE数据段后未使用的空间

ROP链实现了以下功能：

1. 申请RWX内存
2. 将shellcode从RW内存处拷贝到RWX内存储
3. 执行

注：

在ROP链的构造上，Tal Liberman提供了自己的思路，尽可能简化ROP链，优化思路值得学习

3、恢复执行

注入后需要恢复目标进程的执行，使用未公开的函数ZwContinue

0x04 实际测试

测试系统： Win7 x86

安装python，安装pefile(easy_install pefile)

编译生成AtomBombing.exe、AtomBombingShellcode.exe和AtomBombingShellcode.h

注：

AtomBombingShellcode.h由`\AtomBombingShellcode\Scripts\Post_Link.py`生成，可在AtomBombingShellcode工程中的后期生成事件中查看具体参数，如下

启动chrome.exe，执行AtomBombing.exe，注入成功，如下图

补充：

Windows 8.1 update 3和Windows 10添加了一个新的保护机制CFG(Control Flow Guard),CFG的绕过可参考如下链接：

https://blog.ensilo.com/atombombing-cfg-protected-processes

0x05 利用分析

综合公开资料和实际测试，可以将AtomBombing理解为一个APC注入的升级版： 利用Atom Table传递shellcode，通过NtQueueApcThread实现APC注入，shellcode采用构造ROP链的方式，实现了申请内存、写入payload(弹出计算器)并执行的功能

Atom Table支持Windows全平台，并且短期内该功能不会被取消，也不存在修复措施，所以可以在某种程度上理解为不存在修复AtomBombing的补丁

但是，想实现AtomBombing的利用，需要综合考虑多个问题(如获取处于警戒状态的线程、通过NtQueueApcThread传入参数、寻找RX内存，构造ROP链等)，利用门槛较

并不适用于所有进程(目标进程中至少有一个线程处于警戒状态或者能够进入警戒状态)

能绕过部分杀毒软件，但不能绕过所有的杀毒软件(使用NtQueueApcThread进行注入)

0x06 检测防御

将AtomBombing理解为APC注入的升级版，所以参照APC注入的防御方法即可，攻击者首先需要获得系统的执行权限，并找到符合条件的进程

检测：

监控NtQueueApcThread函数的调用

0x07 小结

本文介绍了AtomBombing的实现思路和关键技术，经过实际测试，得出最终结论，AtomBombing是一种新的DLL注入方法，可以理解为一个APC注入的升级版：利用Atom
Table传递shellcode，通过NtQueueApcThread实现APC注入，shellcode采用构造ROP链的方式，实现了申请内存、写入payload(弹出计算器)并执行的功能

上一篇：渗透技巧——利用netsh抓取连接... 下一篇：漏洞视频征集活动规则

1. 0 条回复
   - 动动手指，沙发就是你的了！