HBGary
DETECT. DIAGNOSE. RESPOND.

Fingerprinting Malware Authors

Introductory Case Study of a Chinese APT

HBGary
DETECT. DIAGNOSE. RESPOND.

The Bad Guys are Winning

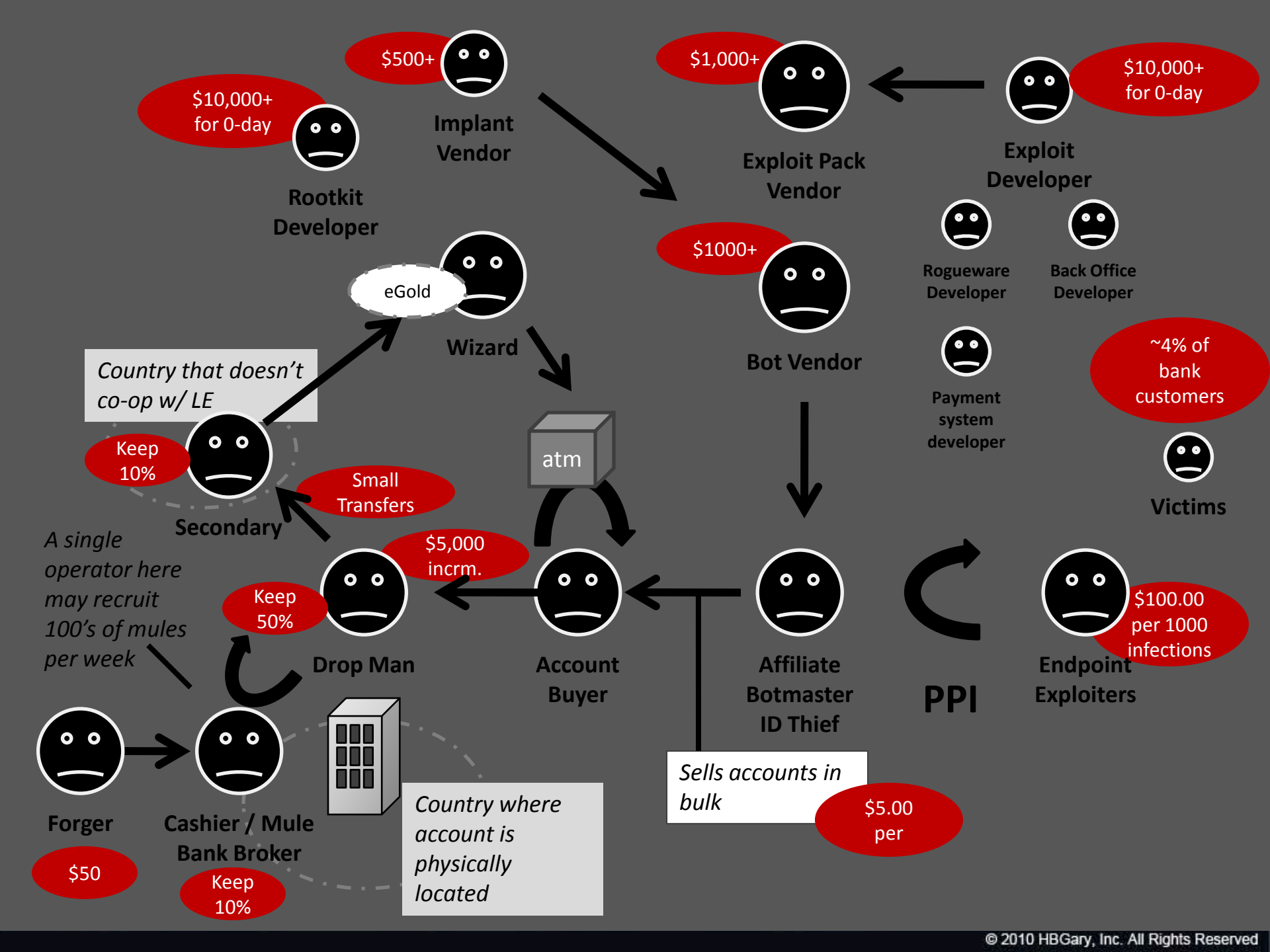
- Cybercrime & espionage is the dominant criminal problem globally, surpassing the drug trade
  - Russians made more money last year in banking fraud than the Columbians made selling cocaine
  - Chinese are crawling all over commercial & government networks
- The largest computing cloud in the world is controlled by Conficker
  - 6.4 million computer systems*
  - 230 countries
  - 230 top level domains globally
  - 18 million+ CPUs
  - 28 terabits per second of bandwidth

*http://www.readwriteweb.com/cloud/2010/04/the-largest-cloud-in-the-world.php

HBGary
DETECT. DIAGNOSE. RESPOND.

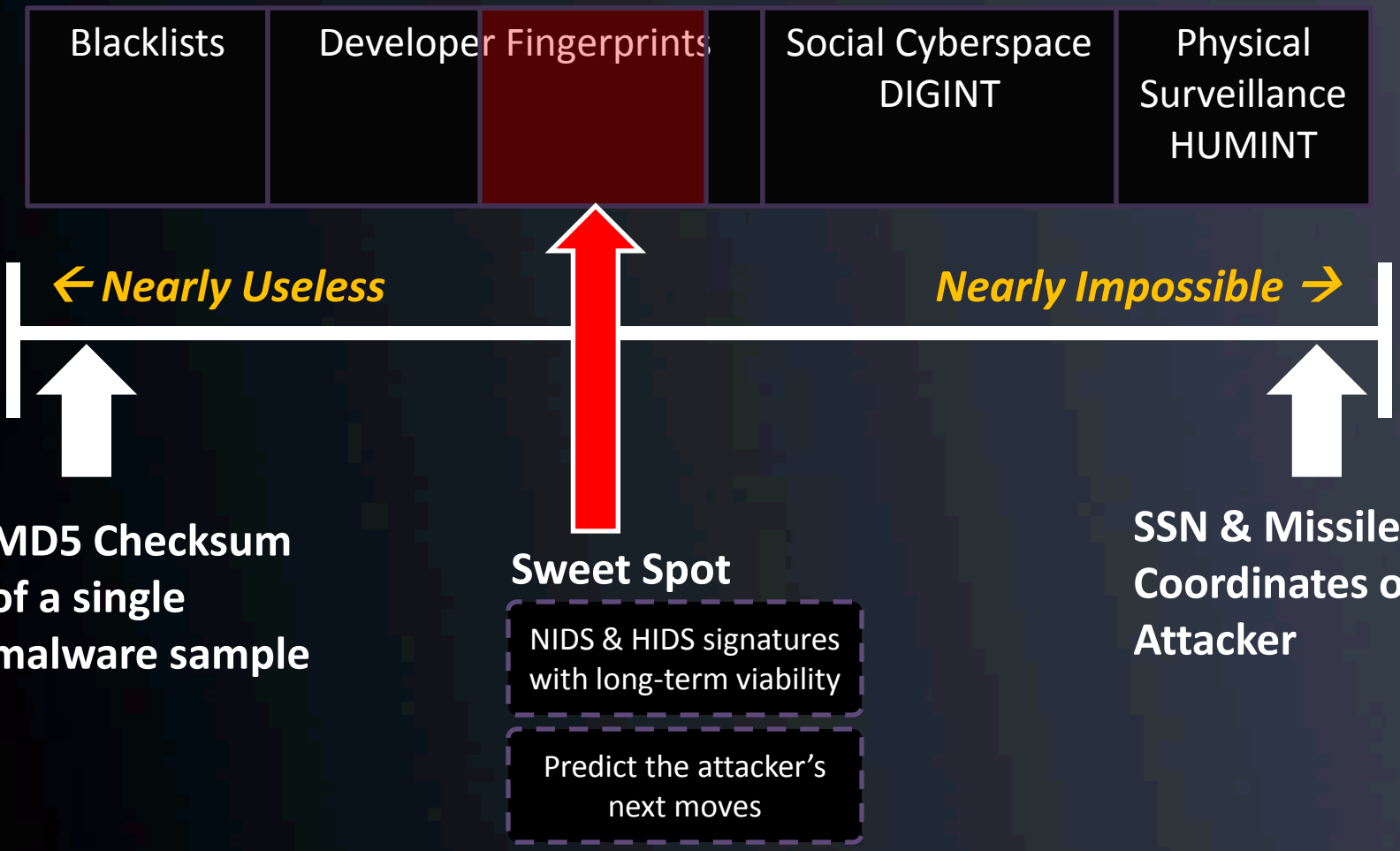
Humans

- Attribution is about the human behind the malware, not the specific malware variants
- Focus must be on human-influenced factors



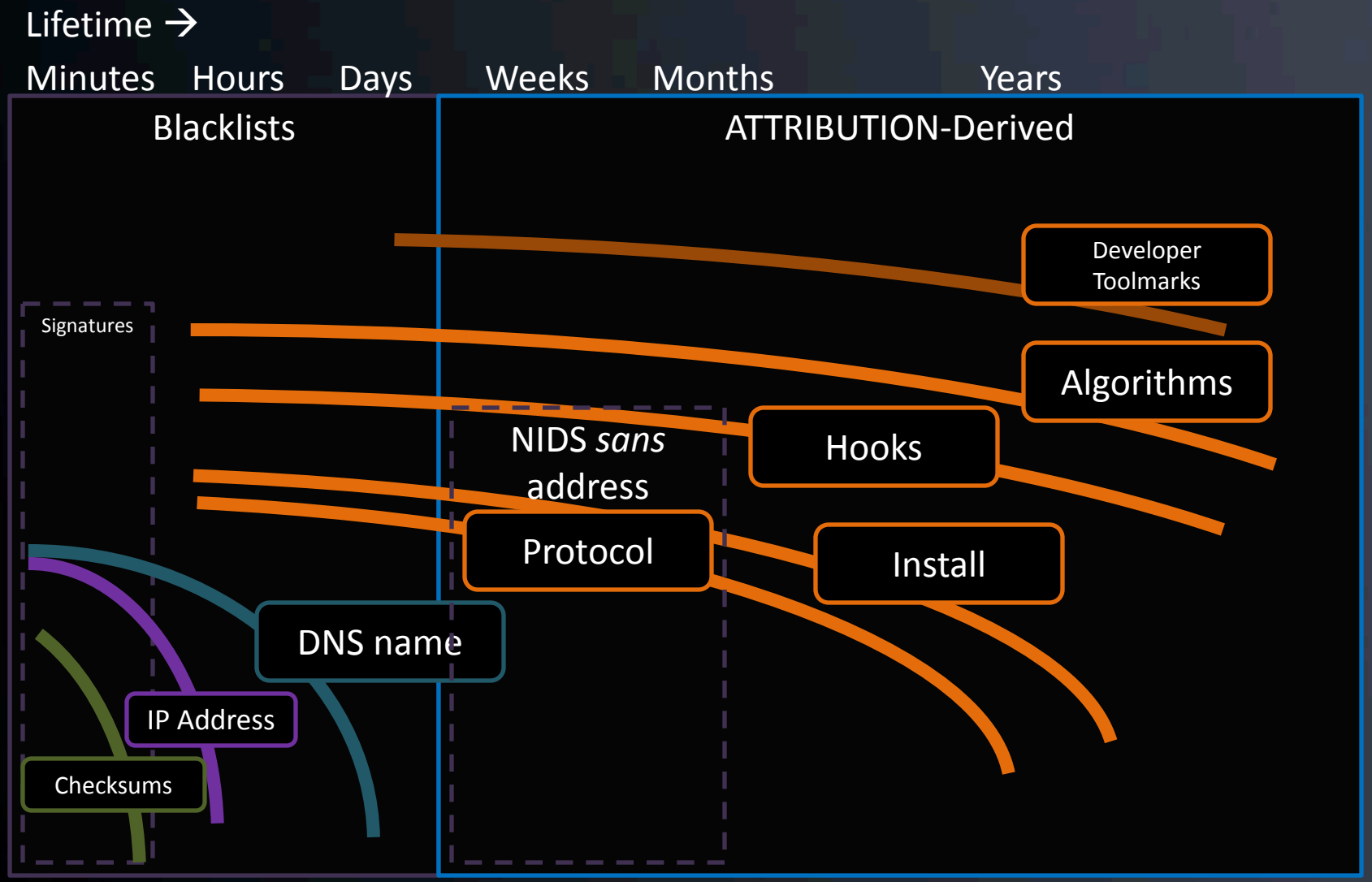
HBGary
DETECT. DIAGNOSE. RESPOND.

Intelligence Spectrum



HBGary
DETECT. DIAGNOSE. RESPOND.

Intel Value Window



HBGary
DETECT. DIAGNOSE. RESPOND.

Rule #1

- The human is lazy
  - The use kits and systems to change checksums, hide from A/V, and get around IDS
  - They DON'T rewrite their code every morning

HBGary
DETECT. DIAGNOSE. RESPOND.

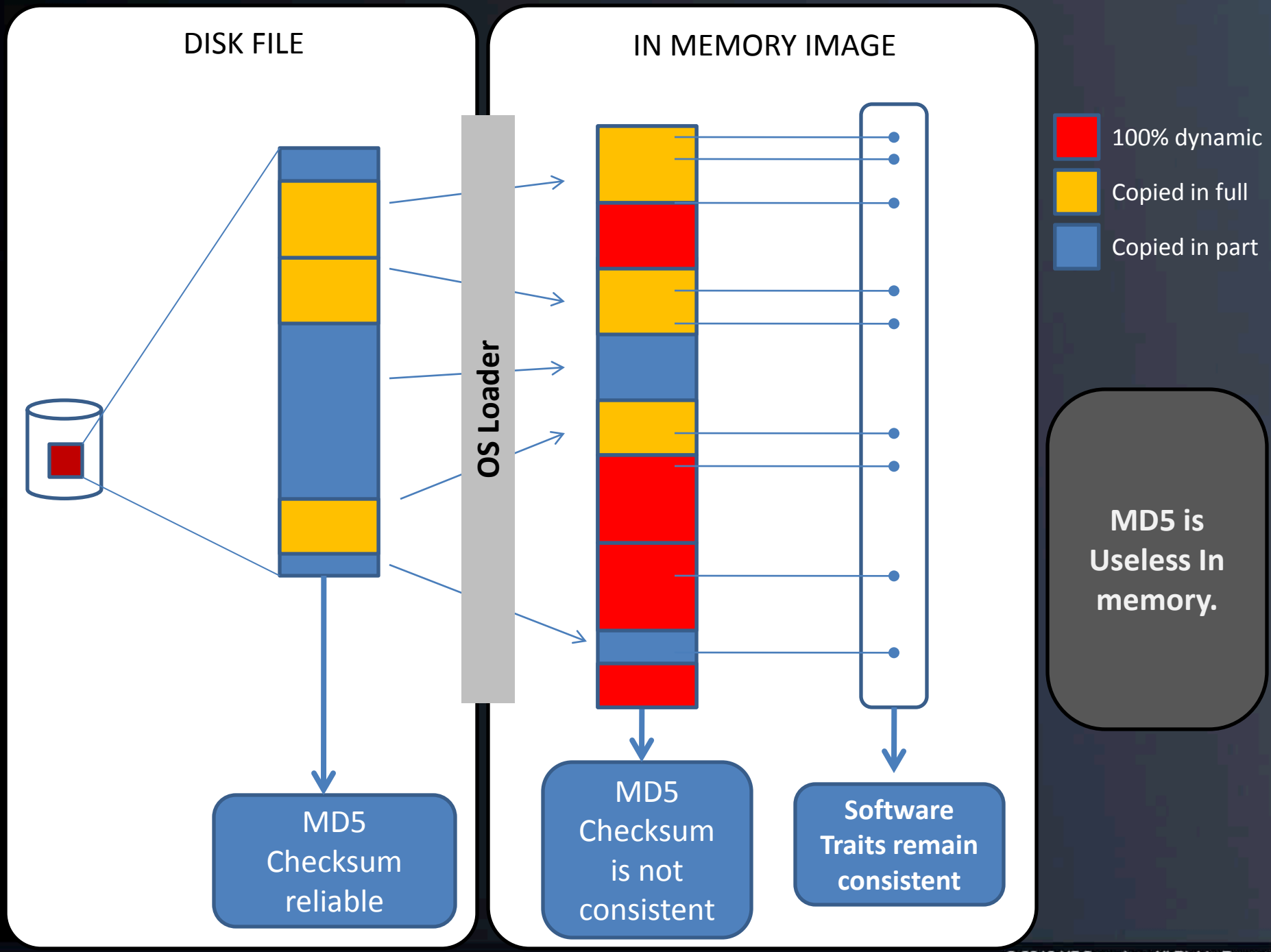
Rule #2

- Most attackers are focused on rapid reaction to network-level filtering and black-holes
  - Multiple DynDNS C2 servers, multiple C2 protocols, obfuscation of network traffic
- They are not-so-focused on host level stealth
  - Most malware is simple in nature, and works great
  - Enterprises rely on A/V for host, and A/V doesn't work, and the attackers know this

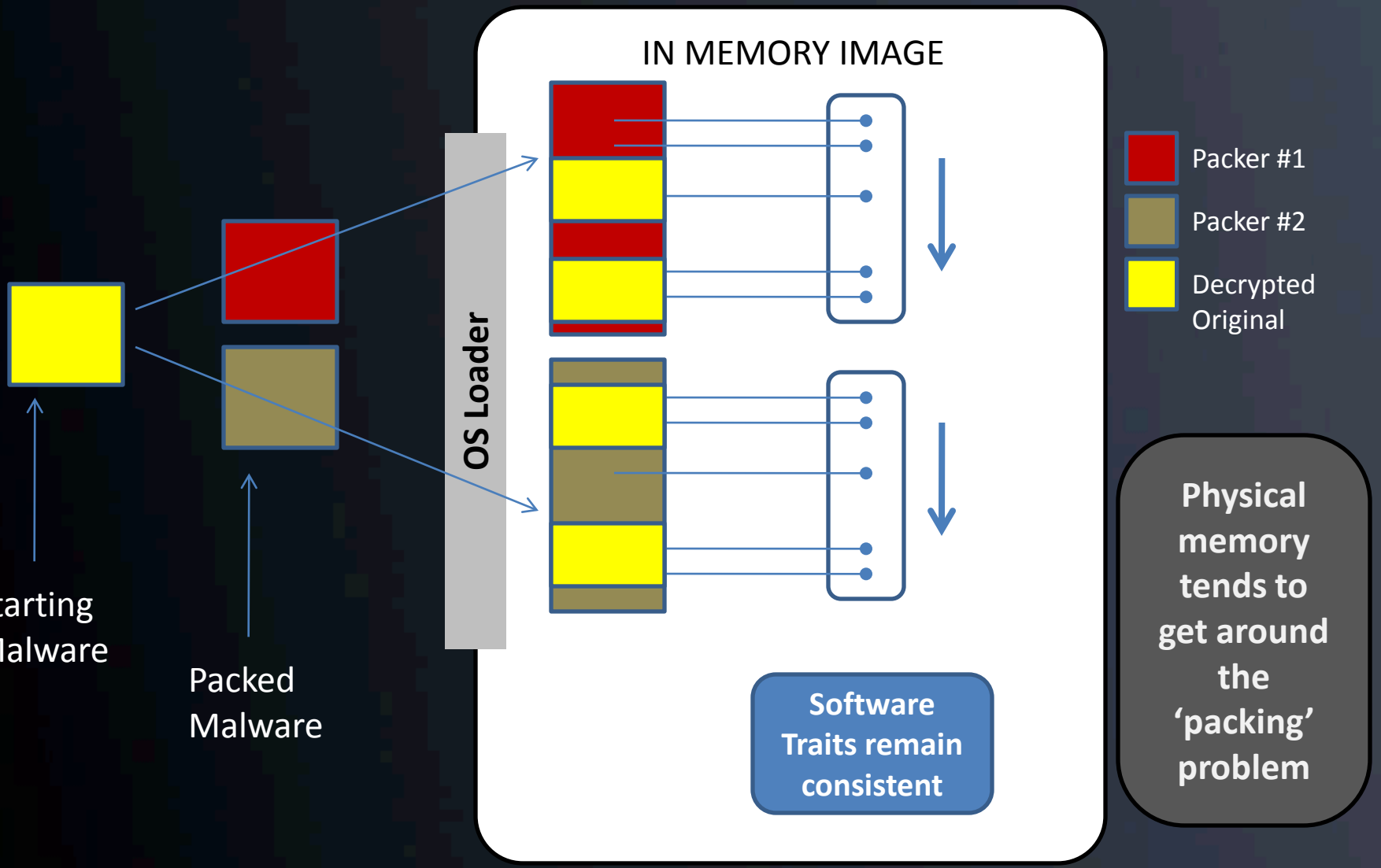
HBGary
DETECT. DIAGNOSE. RESPOND.

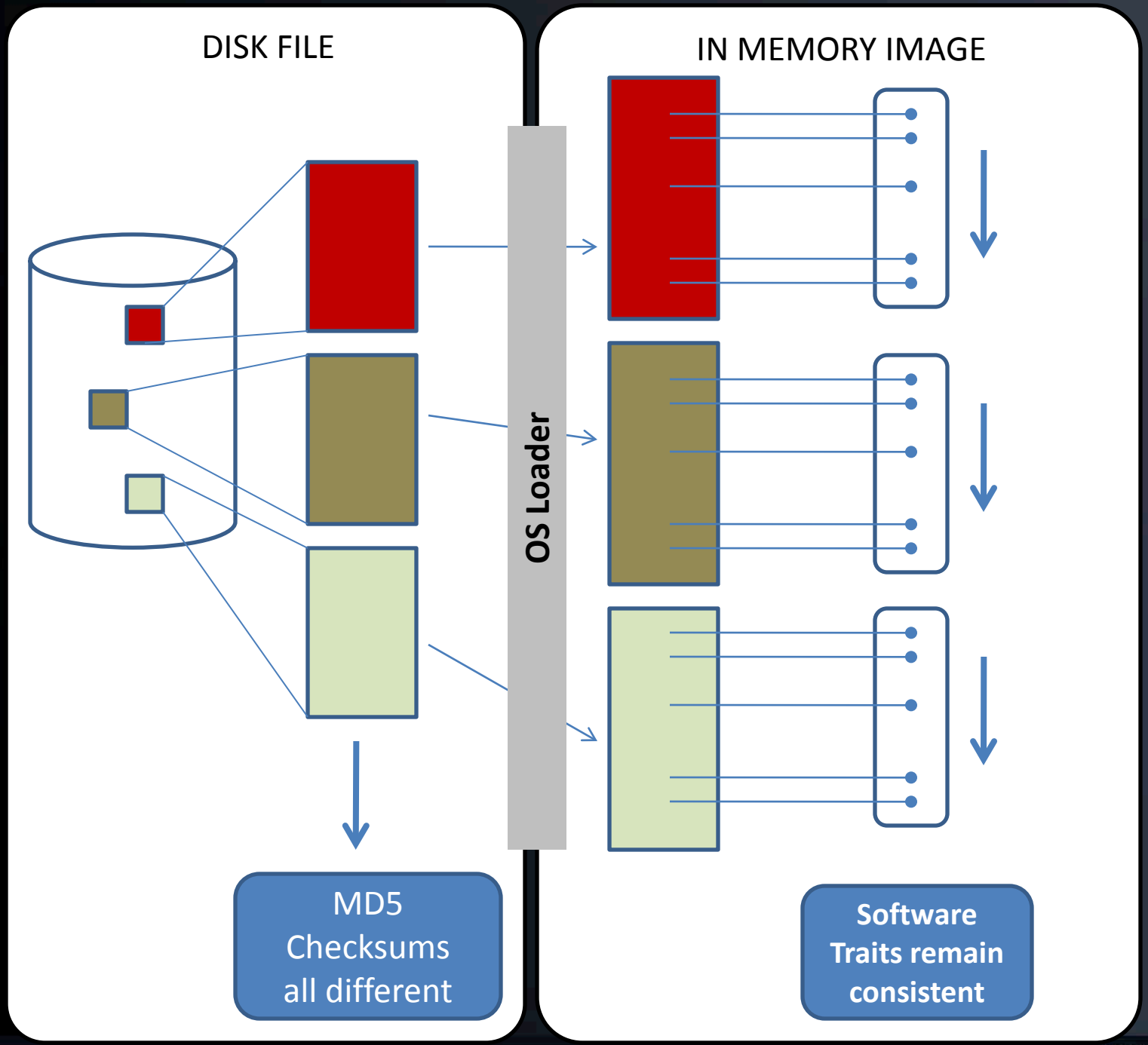
Rule #3

- Physical memory is King
  - Once executing in memory, code has to be revealed, data has to be decrypted



HBGary
DETECT. DIAGNOSE. RESPOND.





Same malware compiled in three different ways

HBGary
DETECT. DIAGNOSE. RESPOND.

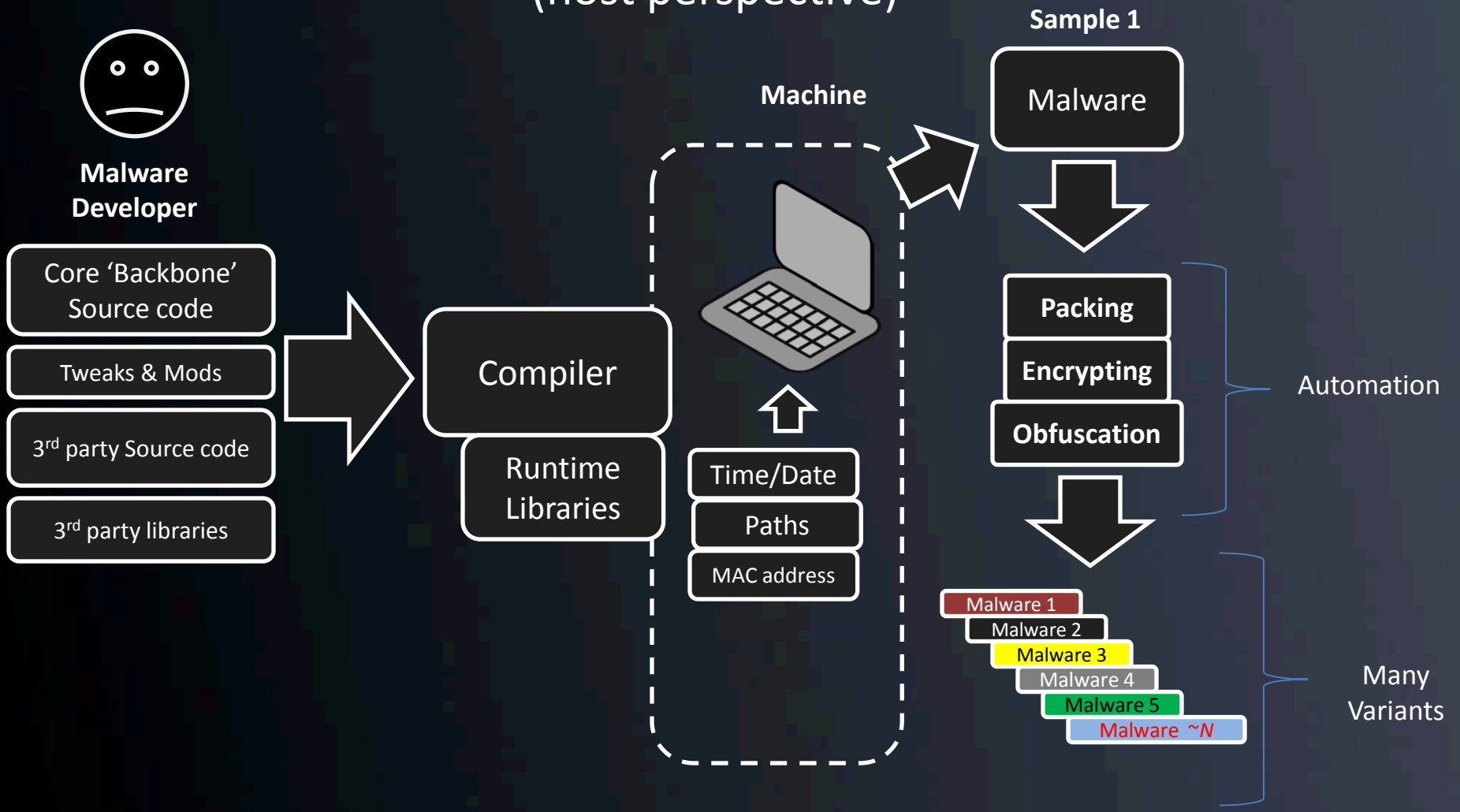
Attribution is Not Hard

- If you can read a packet sniffer, you can attribute malware
  - Yes, this means more people in your organization can do this
  - Focus on strings and human-readable data within a malware program
  - In most cases, code-level reverse engineering is **not required**

HBGary
DETECT. DIAGNOSE. RESPOND.

The Flow of Forensic Toolmarks

(host perspective)

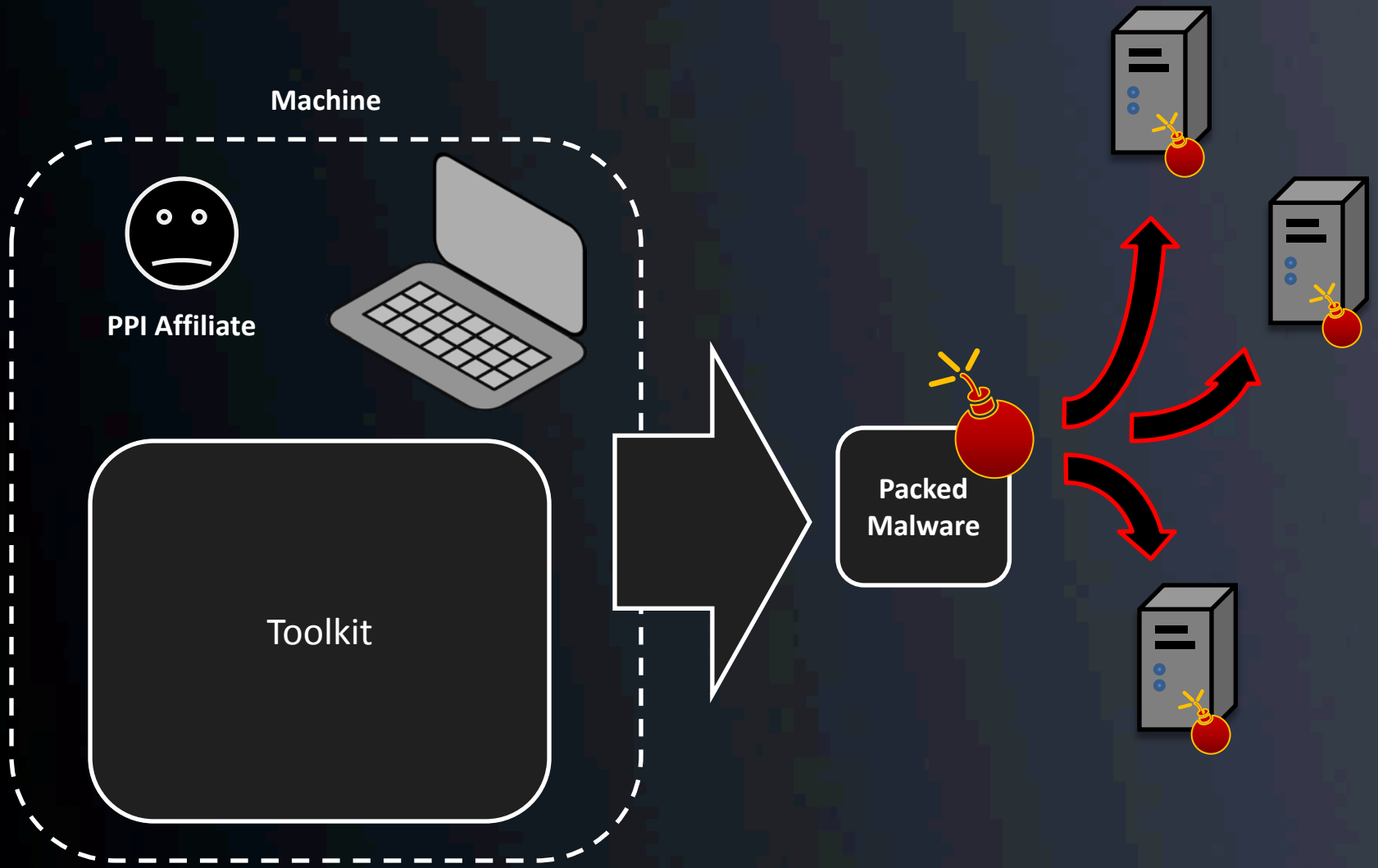


HBGary
DETECT. DIAGNOSE. RESPOND.

Developer Fingerprints

HBGary
DETECT. DIAGNOSE. RESPOND.

Toolkit Fingerprints

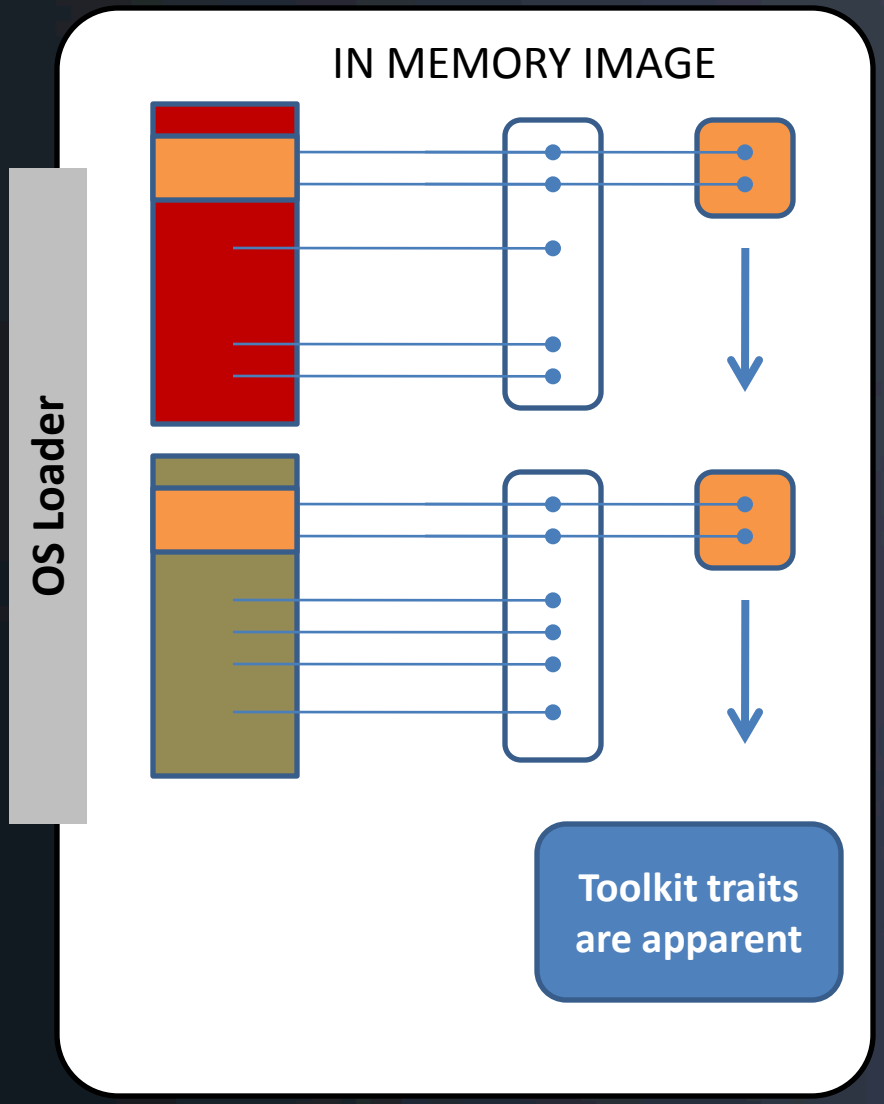


HBGary
DETECT. DIAGNOSE. RESPOND.

Malware Tookit

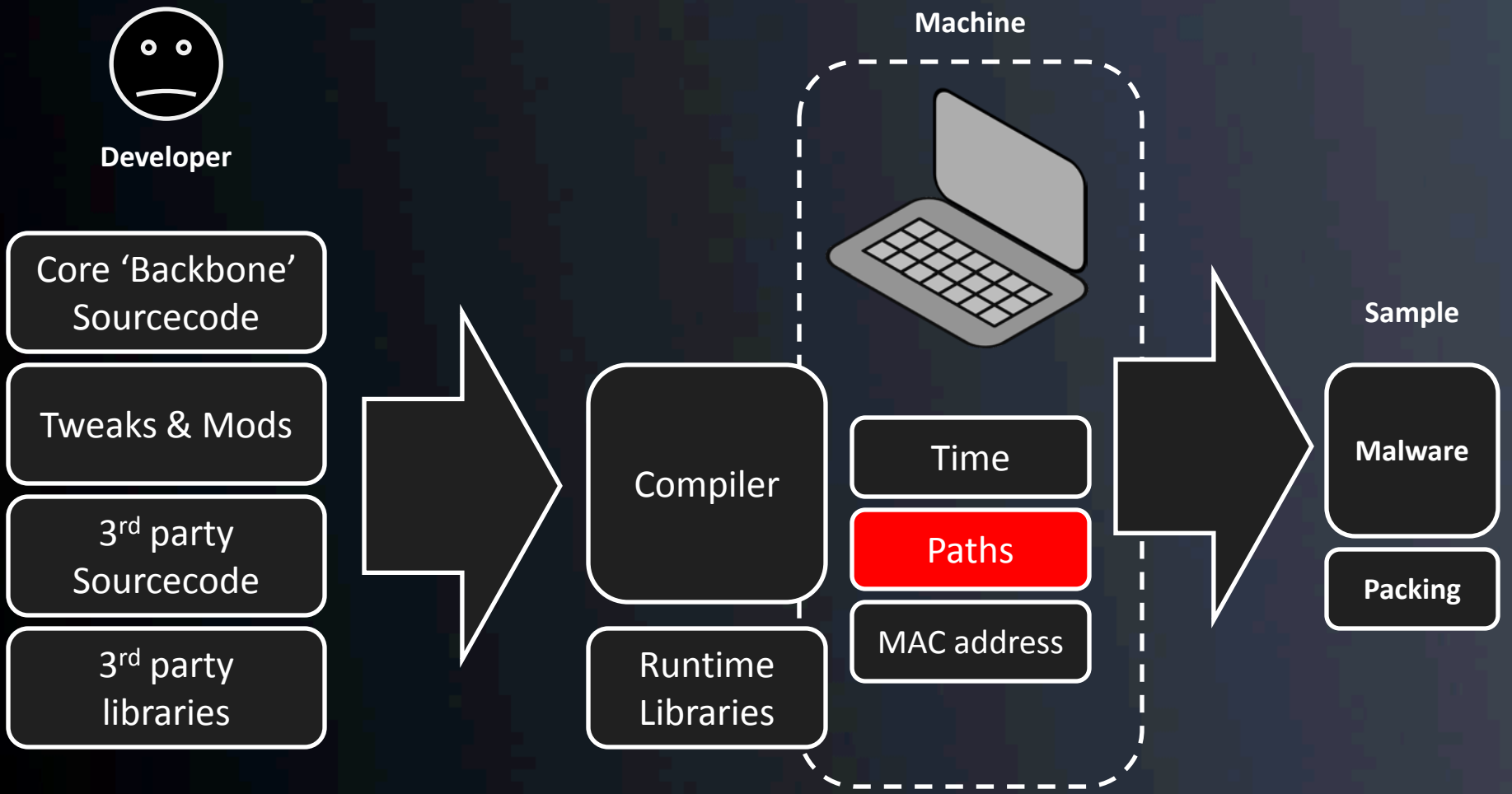
Different Malware Authors Using Same Toolkit

Packed



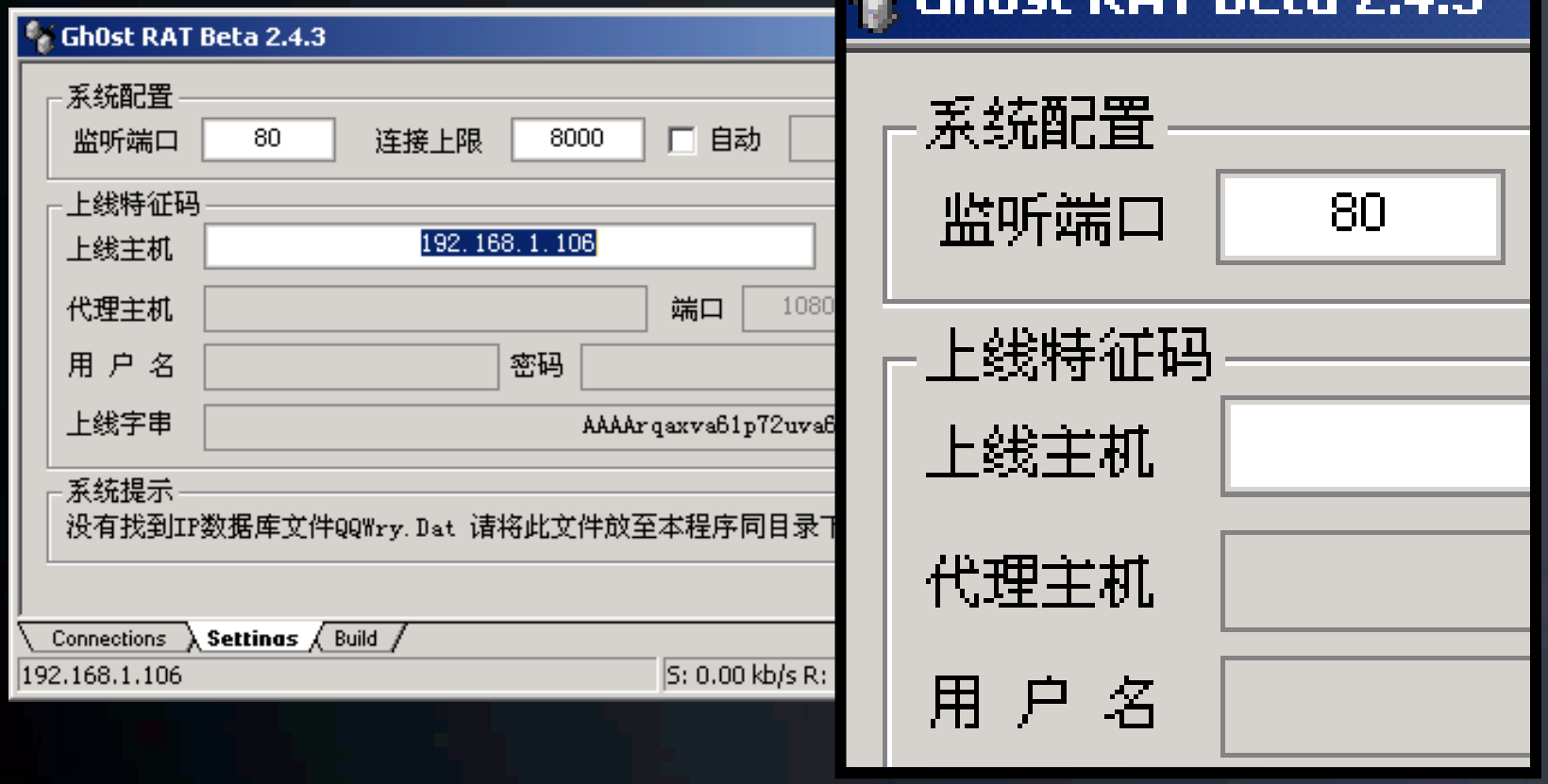
Toolkits can be detected

Paths



HBGary
DETECT. DIAGNOSE. RESPOND.

Example: Gh0stNet



GhostNet: Dropper

UPX!

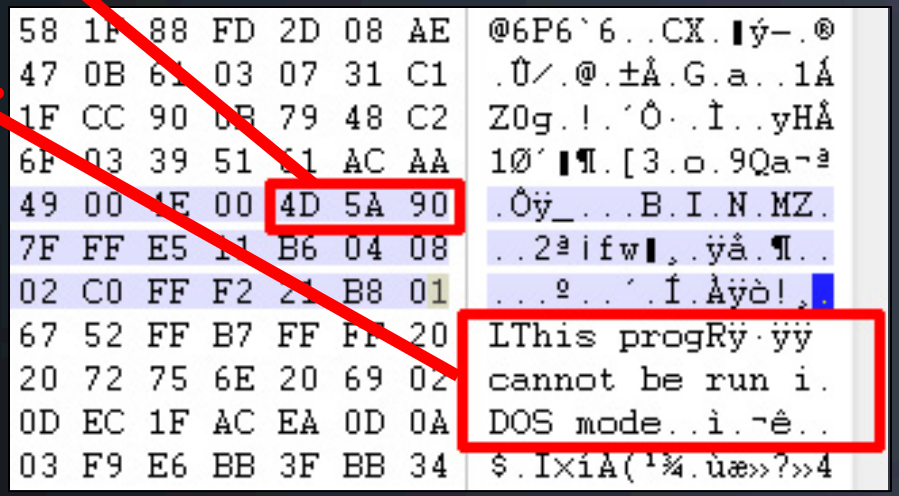
¶üÿÿU‹ìƒìSVW3ÿÿ

Packer Signature

MZx90

This progRy. y cannot be run in DOS mode

Embedded executable
NOTE: Packing is not fully effective here



GhostNet: Dropper

UPX!

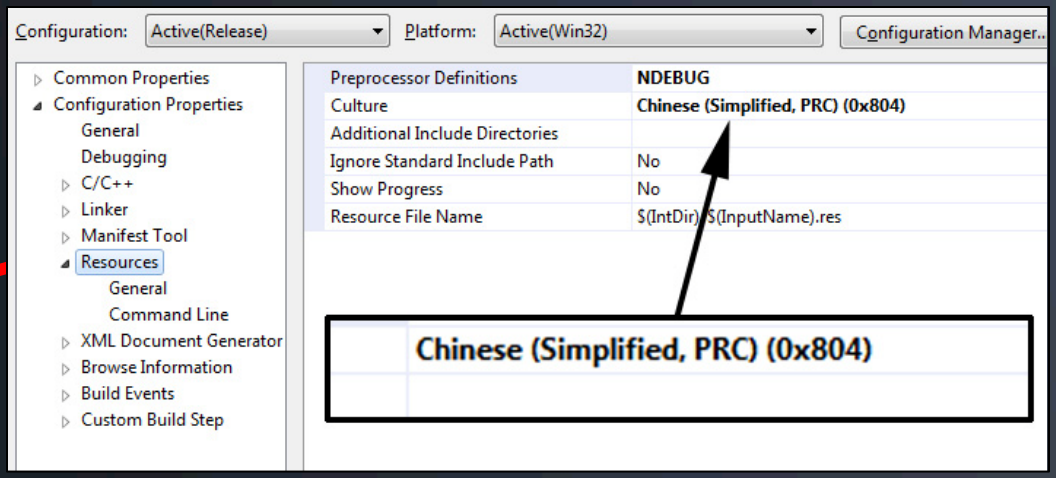
¶üÿÿU‹ìƒìSVW3ÿÿ

Resource Culture Code

0x0804

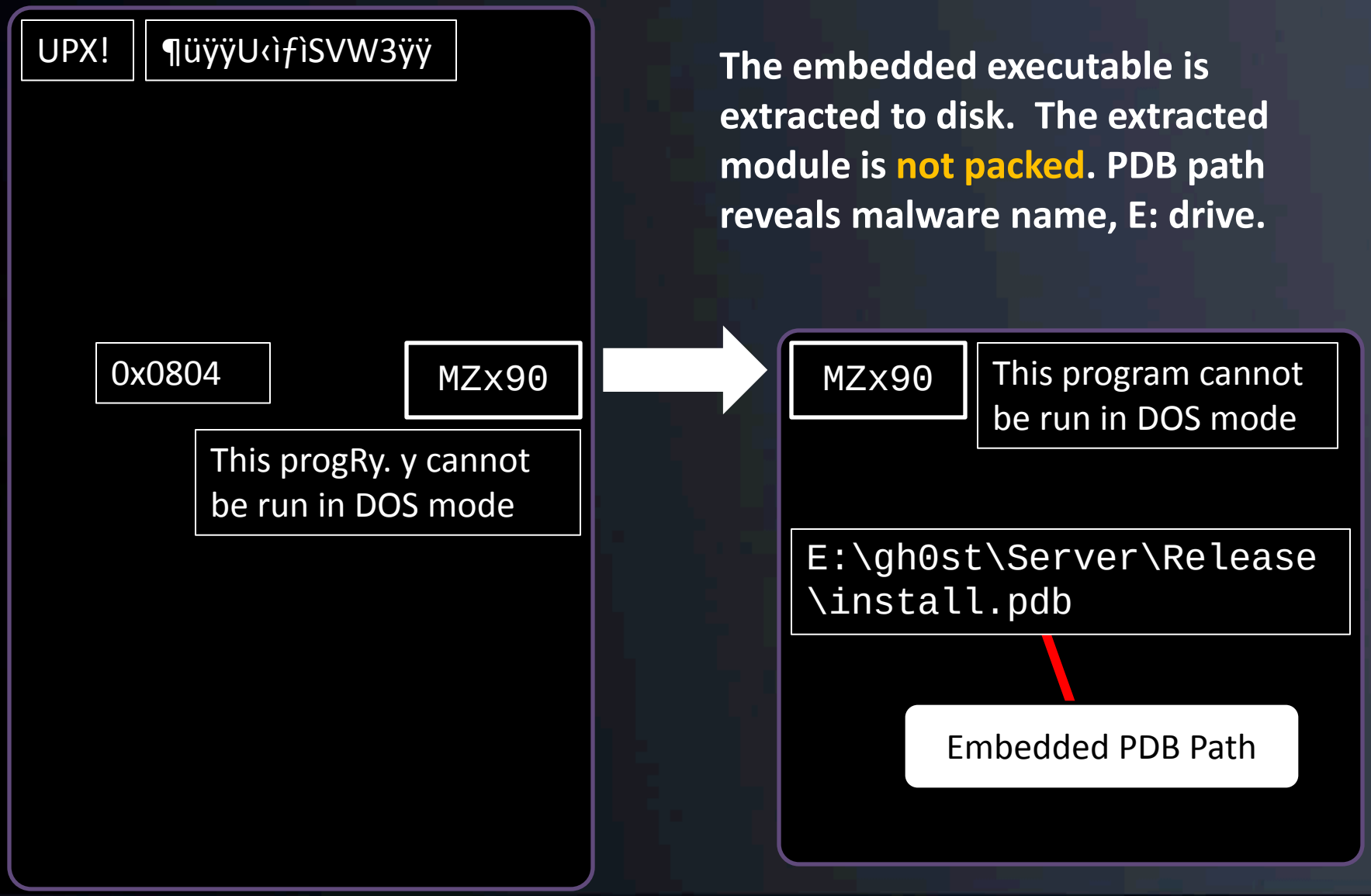
MZx90

This progRy. y cannot be run in DOS mode



The embedded executable is tagged with Chinese PRC Culture code

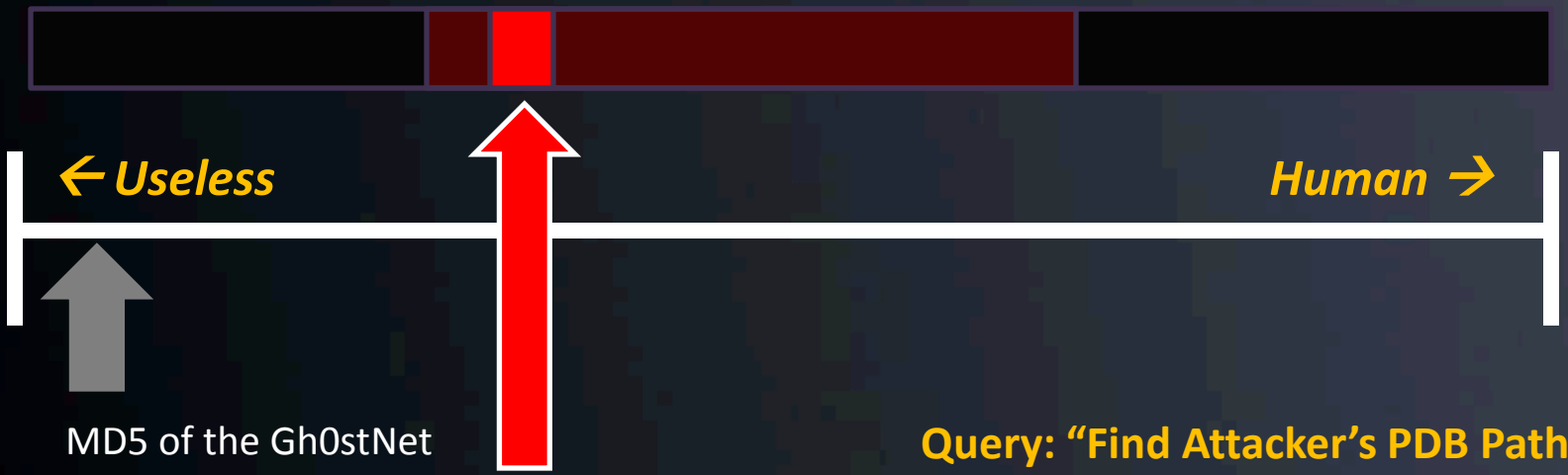
GhostNet: Dropper



The embedded executable is extracted to disk. The extracted module is not packed. PDB path reveals malware name, E: drive.

HBGary
DETECT. DIAGNOSE. RESPOND.

For Immediate Defense…



Query: "Find Attacker's PDB Path"

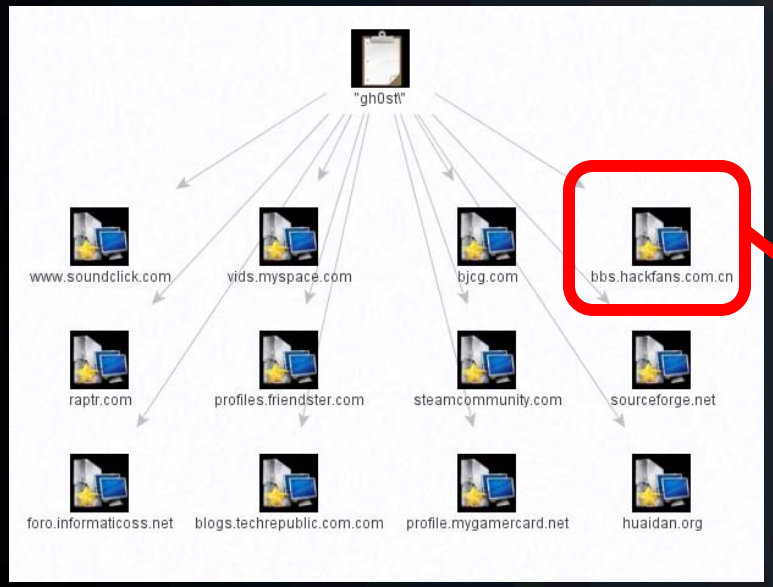
RawVolume.File.BinaryData

contains

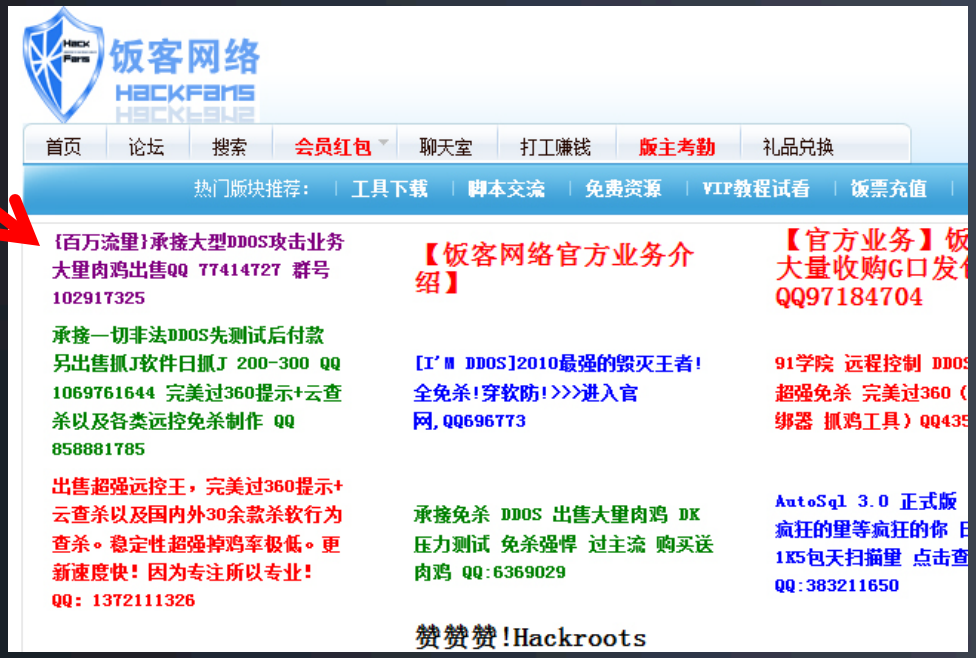
"gh0st\"

Link Analysis

“gh0st\”

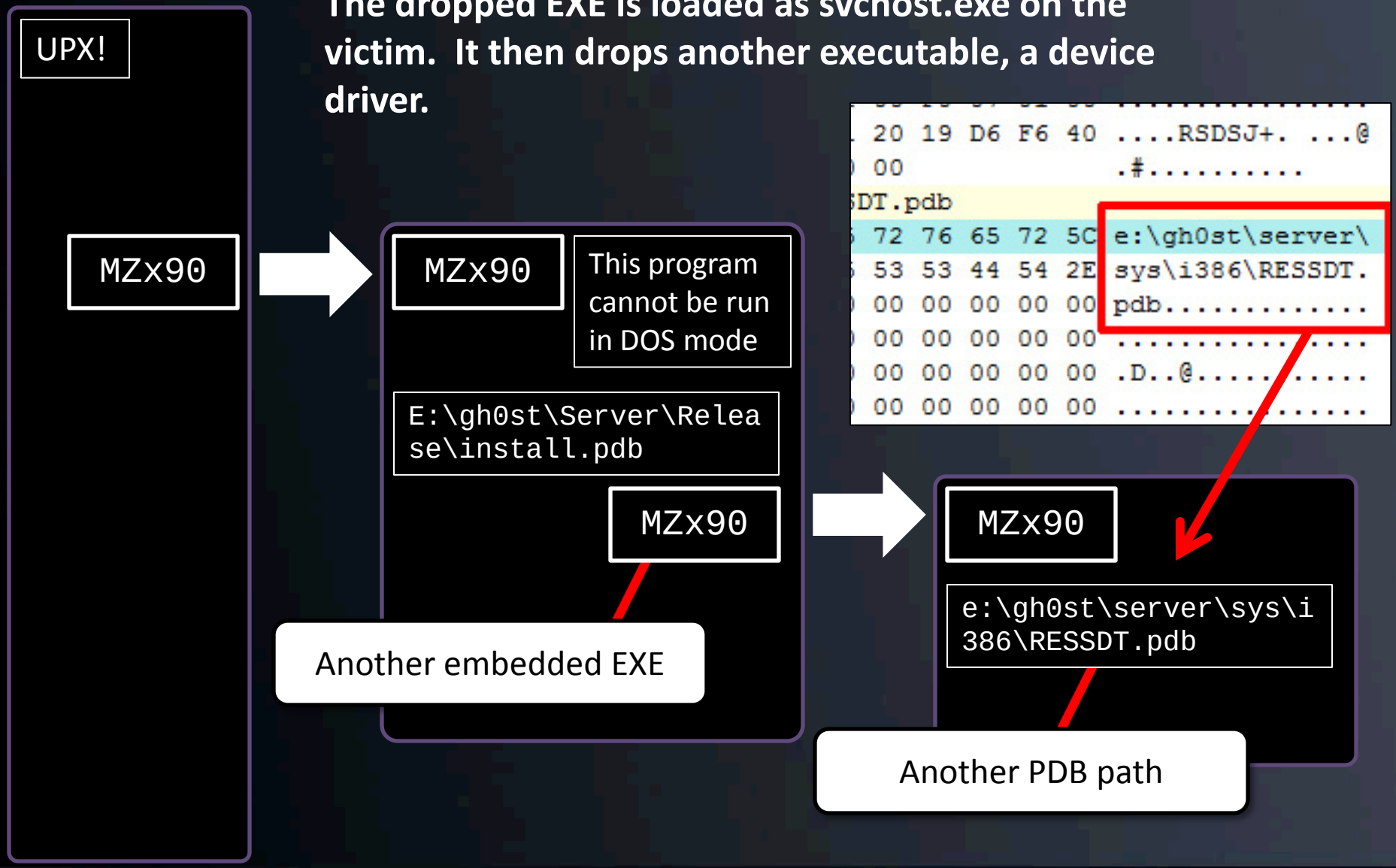


The web reveals Chinese hacker sites that reference the “gh0st\” artifact



GhostNet: Backdoor

The dropped EXE is loaded as svchost.exe on the victim. It then drops another executable, a device driver.



HBGary
DETECT. DIAGNOSE. RESPOND.

Our defense...

Query: "Find Attacker's PDB Path"

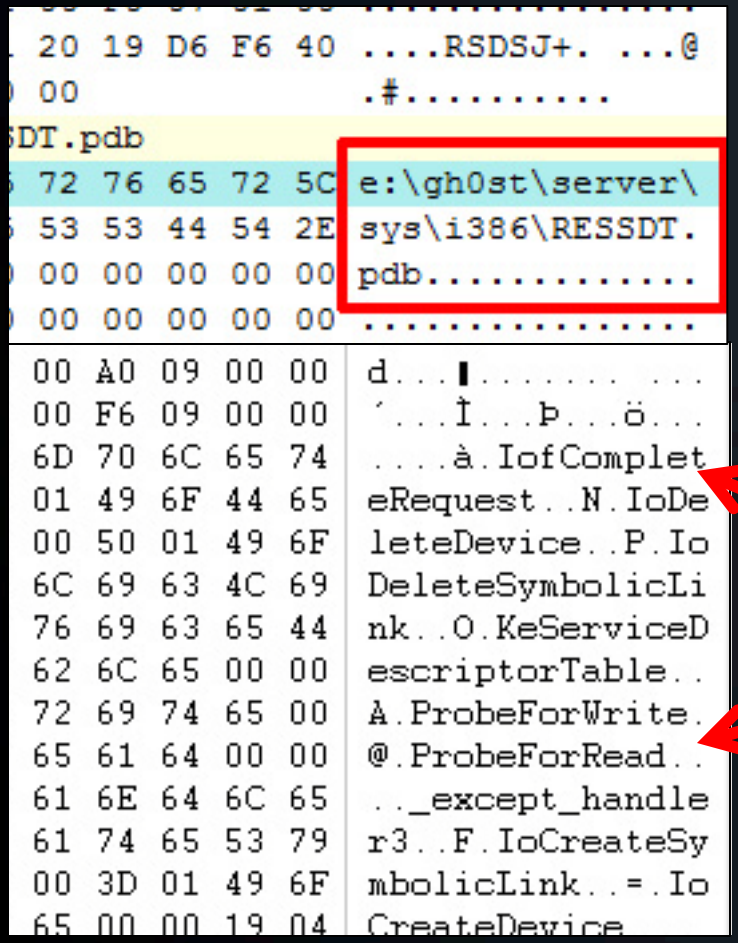
RawVolume.File.BinaryData

contains

`"gh0st\"`

Even if we had not known about the second executable, our defense would have worked. This is how moving towards the human offers predicative capability.

What do we know...



i386 directory is common to device drivers. Other clues:

1. sys directory
2. 'SSDT' in the name

SSDT means System Service Descriptor Table – this is a common place for rootkits and HIPS products to place hooks.

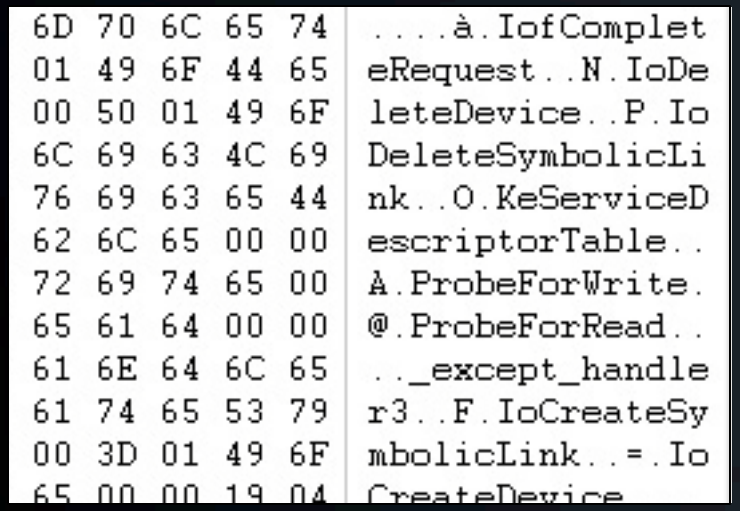
Also, embedded strings in the binary are known driver calls:

1. IoXXXX family
2. KeServiceDescriptorTable
3. ProbeForXXXX

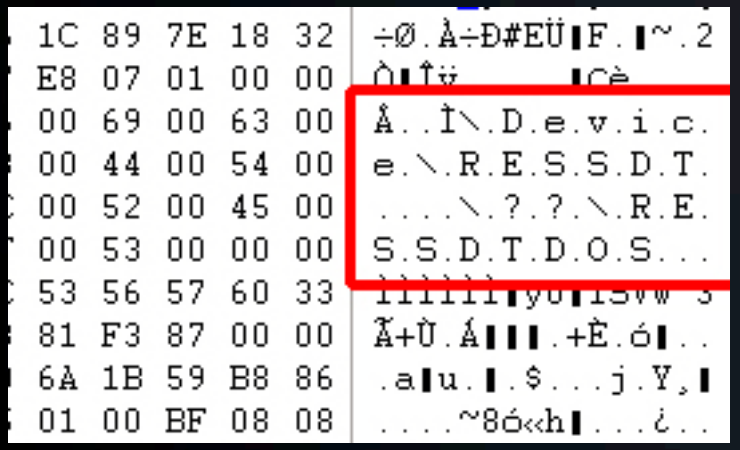
KeServiceDescriptorTable is used when SSDT hooks are placed. We know this is a hooker.

HBGary
DETECT. DIAGNOSE. RESPOND.

What do we know...



```
6D 70 6C 65 74   ....à.IofComplet
01 49 6F 44 65   eRequest..N.IoDe
00 50 01 49 6F   leteDevice..P.Io
6C 69 63 4C 69   DeleteSymbolicLi
76 69 63 65 44   nk..O.KeServiceD
62 6C 65 00 00   escriptorTable..
72 69 74 65 00   A.ProbeForWrite.
65 61 64 00 00   @.ProbeForRead..
61 6E 64 6C 65   ..._except_handle
61 74 65 53 79   r3..F.IoCreateSy
00 3D 01 49 6F   mbolicLink..=.Io
65 00 00 19 04   CreateDevice
```



```
1C 89 7E 18 32   ÷Ø.À÷Ð#EÜ▮F.▮~.2
E8 07 01 00 00   Ò▮Îü      ▮Cè
00 69 00 63 00   Å..Ì\.D.e.v.i.c.
00 44 00 54 00   e.\.R.E.S.S.D.T.
00 52 00 45 00   ....\.?.?.\.R.E.
00 53 00 00 00   S.S.D.T.D.O.S...
53 56 57 60 33   ìììììì▮yu▮1SVW`3
81 F3 87 00 00   Ã+Ù.Á▮▮▮.+È.ó▮..
6A 1B 59 B8 86   .a▮u.▮.$...j.Y¸▮
01 00 BF 08 08   ....~8ó«h▮...¿..
```

IofCompleteRequest, **IoCreateDevice**, **IoCreateSymbolicLink**, and friends are used when the driver communicates to usermode. This means there is a usermode module (a process EXE or DLL) that is used in conjunction with the device driver.

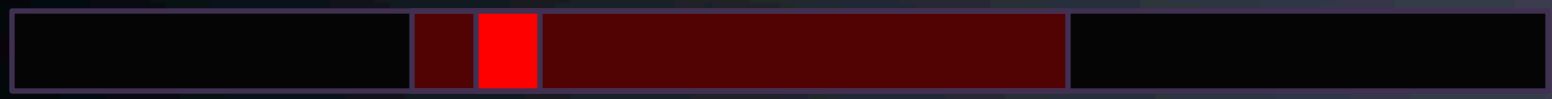
When communication takes place between usermode & kernelmode, there will be a **device path**.

HBGary
DETECT. DIAGNOSE. RESPOND.

For Immediate Defense…

MD5 of the Gh0stNet dropper.EXE

Device Path of the kernel mode driver and the Symbolic Link name



← *Useless* *Human* →

Query: "Find Rootkit Device Path or Symlink"

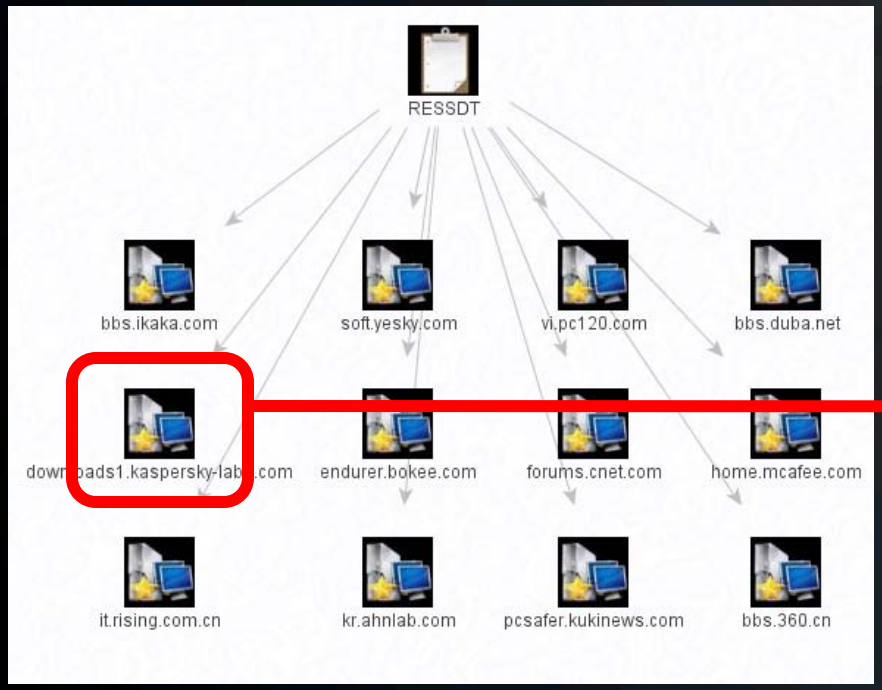
Physmem.WindowsObject.Name

contains

"RESSDT"

Link Analysis

"RESSDT"




```
    Net-Worm.Win32.Rovud.a-c
    Trojan.Win32.ConnectionServices.x-aa
    Worm.Win32.AutoRun.dtx
    Worm.Win32.AutoRun.hr
    Backdoor.Win32.Agent.lad
    not-a-virus:FraudTool.Win32.UltimateDefender.cm
    Trojan-Downloader.Win32.Agent.wbu
    Backdoor.Win32.Small.cyb
    not-a-virus:FraudTool.Win32.XPSecurityCenter.c
    not-a-virus:Downloader.Win32.VistaAntivirus.a
    not-a-virus:FraudTool.Win32.UltimateAntivirus.an
    not-a-virus:FraudTool.Win32.UltimateAntivirus.ap
    Trojan-Spy.Win32.Zbot.dlh
    Trojan-Downloader.Win32.Small.abpz
    Rootkit.Win32.Ressdt.br
    Worm.Win32.AutoRun.lsf
    Worm.Win32.AutoRun.epo
    Worm.Win32.AutoRun.enw
    Backdoor.Win32.UltimateDefender.a
0.0.20 Copyright (C) Kaspersky Lab, Antropov Alexey, Vitaly Kamlu
rved.
**********************************************************************
```

A readme file on Kasperky's site references a Ressdt rootkit.

HBGary
DETECT. DIAGNOSE. RESPOND.

Forensic Toolmarks

e:\gh0st\server\sys\i386\RESSDT.pdb
e:\job\gh0st\Release\Loader.pdb
.?AVCgh0stDoc@@
.?AVCgh0stApp@@
.?AVCgh0stView@@
Cgh0stView
Cgh0stDoc
e:\job\gh0st\Release\gh0st.pdb
C:\gh0st3.6_src\HACKER\i386\HACKE.pdb
\gh0st3.6_src\Server\sys\i386\CHENQI.pdb

Rootkit

Dropper

GUI (MFC)

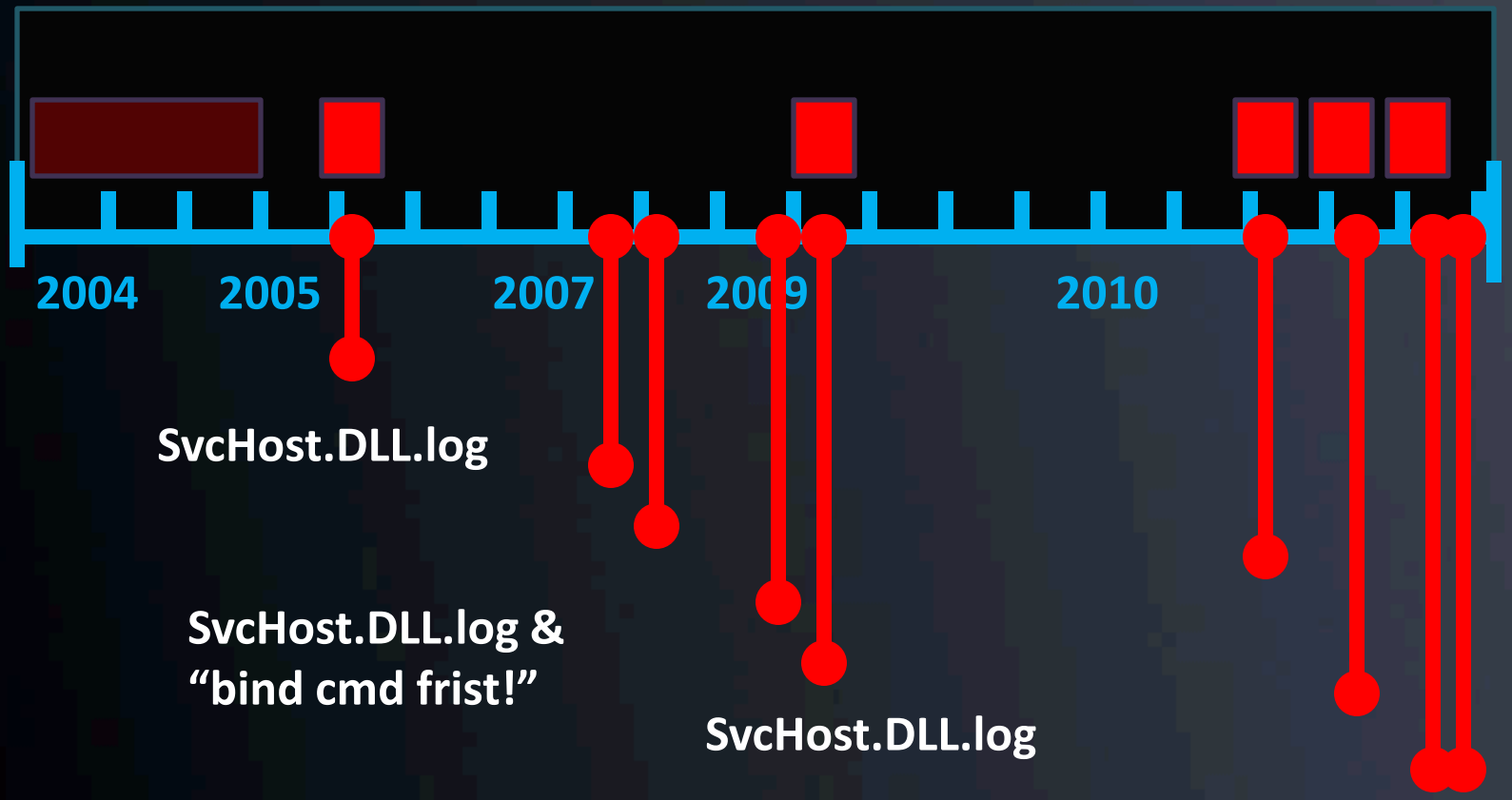
Doc/View is usually MFC

Already at version 3.6

Rootkits

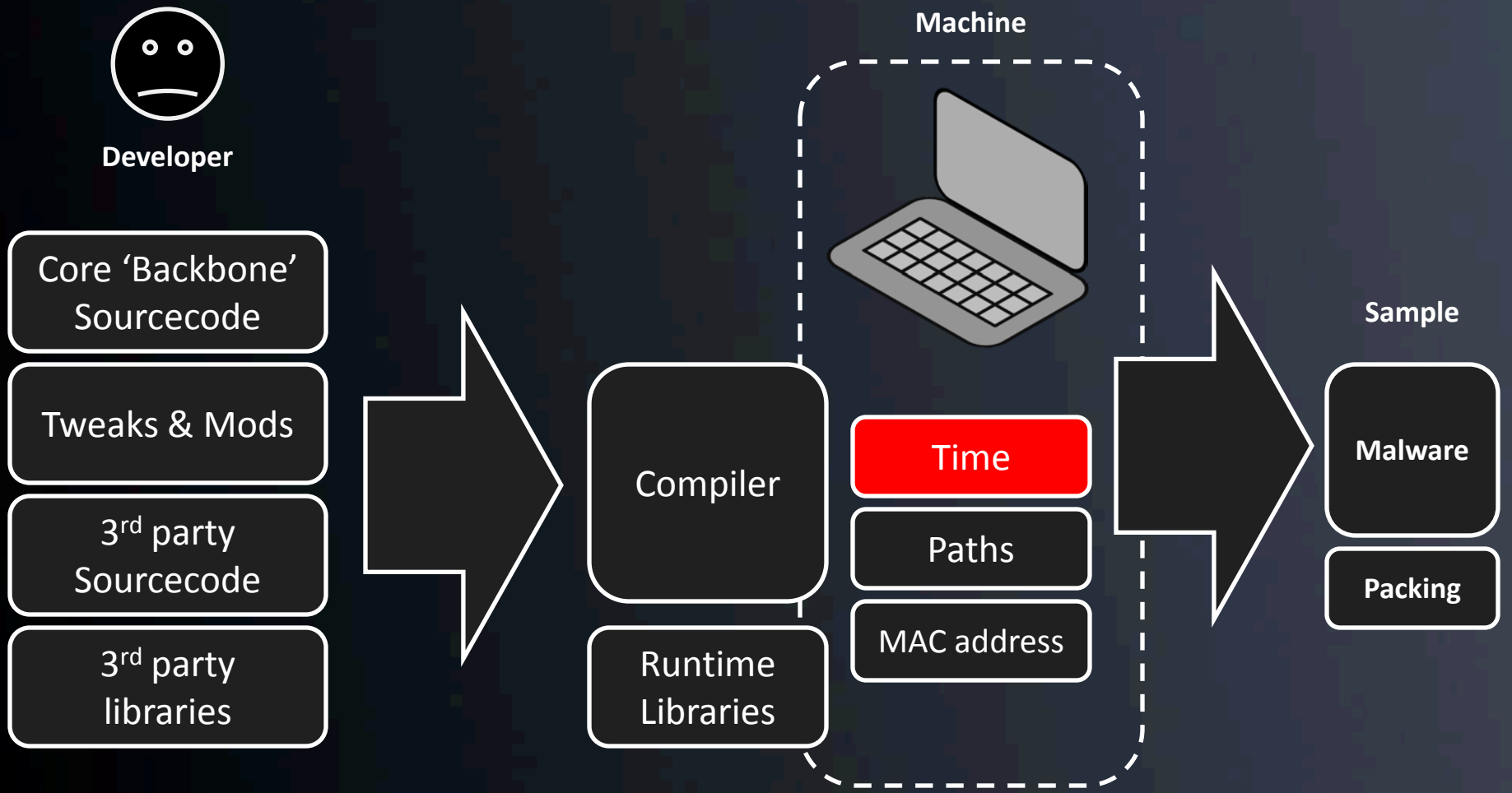
HBGary
DETECT. DIAGNOSE. RESPOND.

Case Study: Chinese APT



HBGary
DETECT. DIAGNOSE. RESPOND.

Timestamps



HBGary
DETECT. DIAGNOSE. RESPOND.

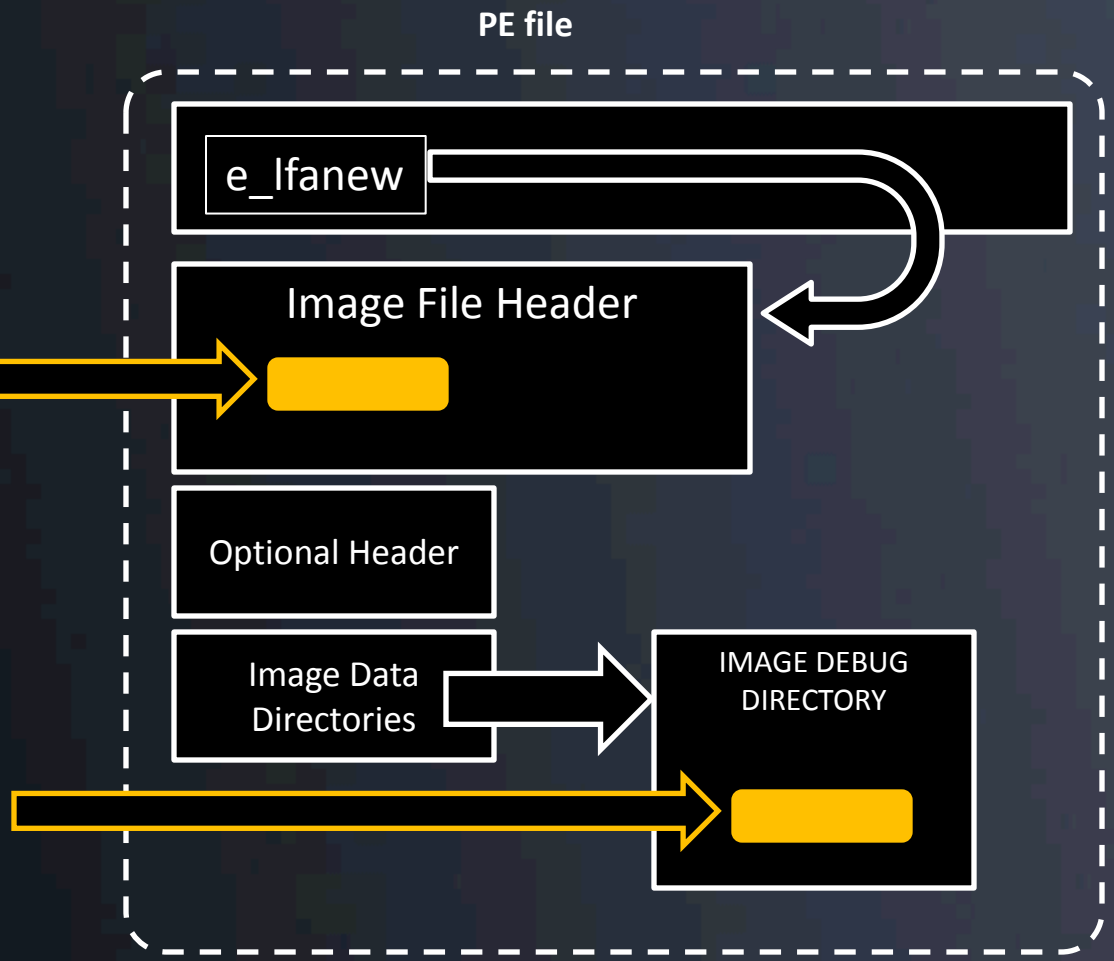
PE Timestamps

Module timestamp*
time_t (32 bit)

The 'lmv' command in WinDBG will show this value..

Debug timestamp
time_t (32 bit)

This is present if an external PDB file is associated with the EXE



*This is not the same as NTFS file times, which are 64 bit and stored in the NTFS file structures.

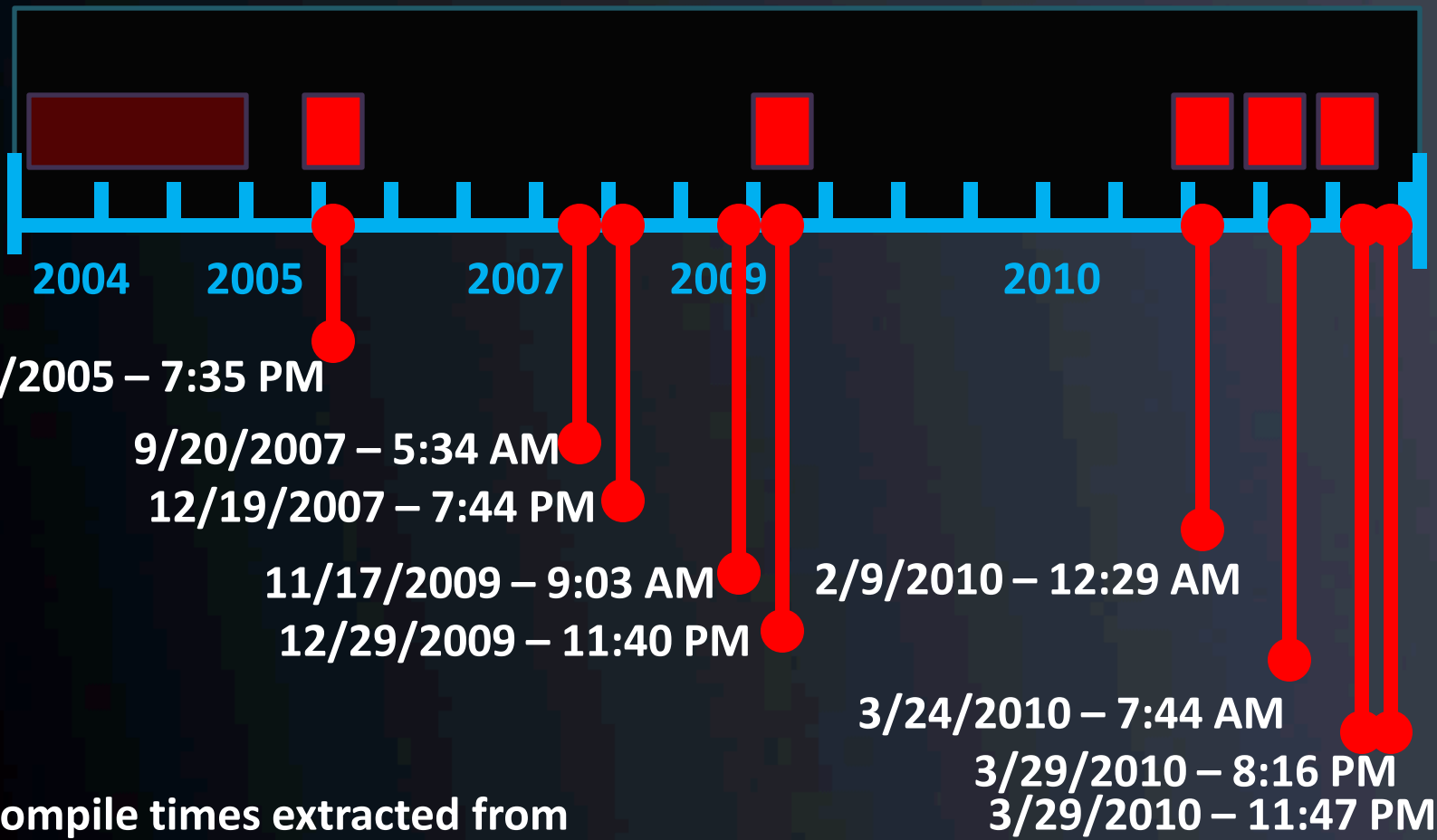
HBGary
DETECT. DIAGNOSE. RESPOND.

Timestamp Formats

- time_t – 32 bit, seconds since Jan. 1 1970 UTC
  - 0x3DE03E0A ← usually start with '3' or '4'
    - '3' started in 1995 and '4' ends in 2012
  - Use 'ctime' function to convert
- FILETIME – 64 bit, 100-nanosecond intervals since Jan. 1 1600 UTC
  - 0x01C195C2.5100E190 ← usually start with '01' and a letter
    - 01A began in 1972 and 01F ends in 2057
  - Use FileTimeToSystemTime(), GetDateFormat(), and GetTimeFormat() to convert

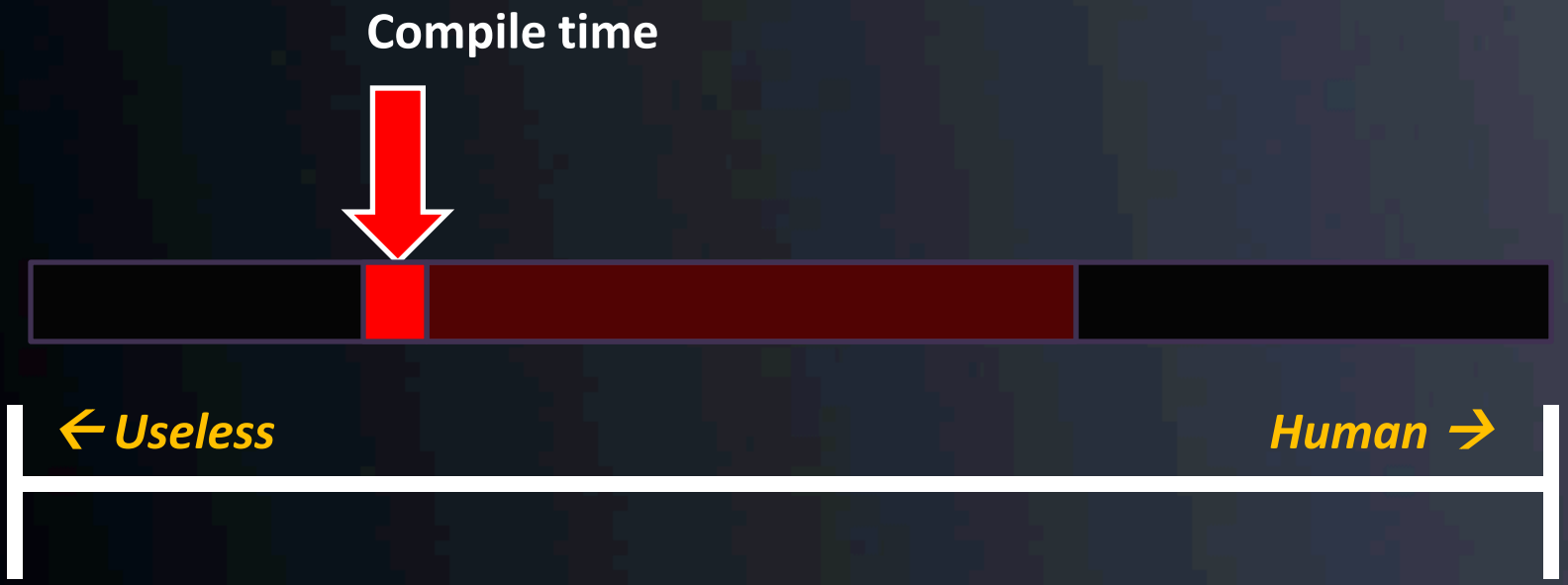
HBGary
DETECT. DIAGNOSE. RESPOND.

Case Study: Chinese APT



Compile times extracted from 'soysauce' backdoor program.

For Immediate Defense…

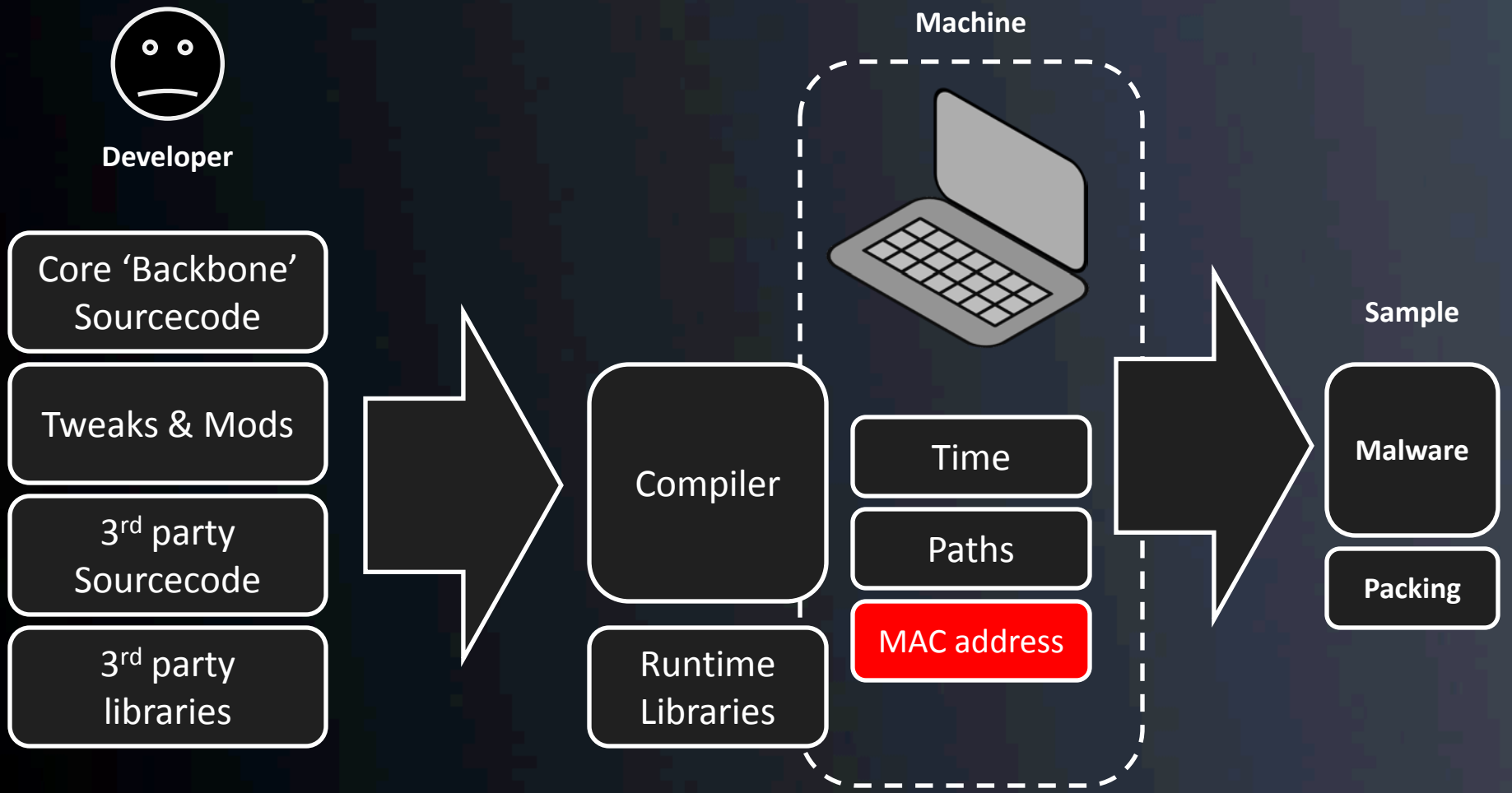


Query: "Find Modules Created Within Attack Window"

RawVolume.File.CompileTime	
>	3/1/2010
<	3/31/2010

HBGary
DETECT. DIAGNOSE. RESPOND.

MAC Address



HBGary
DETECT. DIAGNOSE. RESPOND.

GUID V1

- The OSF specified algorithm for GUID V1 uses the MAC address of the network card for the last 48 bits of the 128 bit GUID
  - This was deprecated on Windows 2000 and greater, so this has limited value

{21EC2020-3AEA-1069-A2DD-08002B30309D}

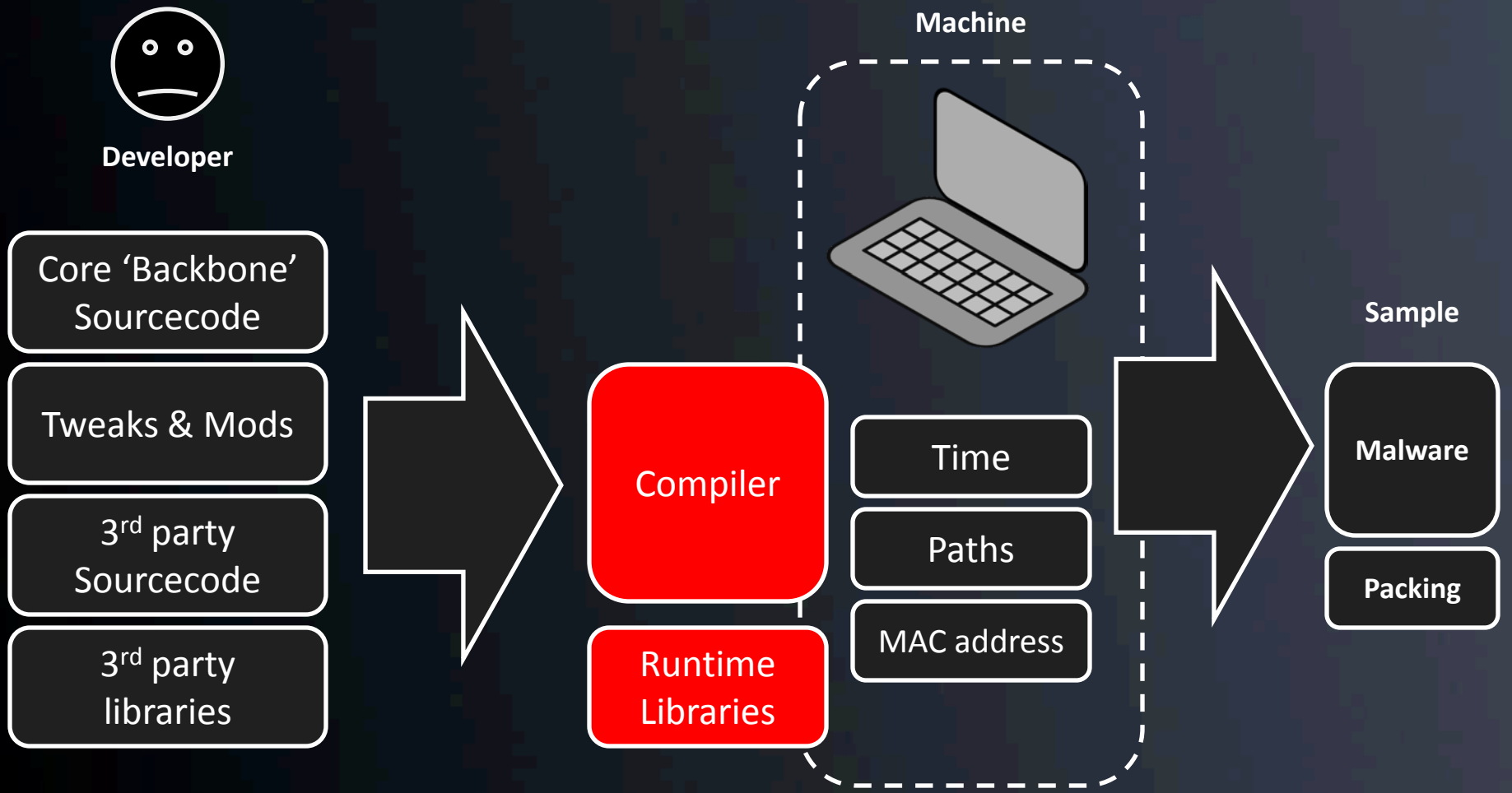
V1 GUIDS have a 1 in this position

This is the MAC of the machine

This technique was used to track the author of the Melissa virus

HBGary
DETECT. DIAGNOSE. RESPOND.

Compiler Version



HBGary
DETECT. DIAGNOSE. RESPOND.

Visual Studio

- Static or dynamic linked runtime library?
- Single-threaded or multi-threaded?
- Use of STL?
- Use of older iostream libraries?*

See: * support.microsoft.com/kb/154753

HBGary
DETECT. DIAGNOSE. RESPOND.

Visual Studio – Static Linking

Version	Libraries linked with	Type	Compiler flag
VC++ .NET 2003 and earlier	LIBC.LIB, LIBCP.LIB	Single Threaded Static	/ML
VC++ .NET 2003 and earlier	LIBCD.LIB, LIBCPD.LIB	Single Threaded Static	/MLd
All	LIBCMT.LIB, LIBCPMT.LIB	Multi-threaded Static	/MT
All	LIBCMTD.LIB, LIBCPMTD.LIB	Multi-threaded Static	/MTd

Visual Studio – Dynamic Linking

Version	DLL Linked with
VC++ 4.2	MSVCRT.DLL/MSVCRTD.DLL
VC++ 5.0	MSVCR50.DLL
VC++ 6.0	MSVCR60.DLL
VC++ .NET 2002	MSVCR70.DLL
VC++ .NET 2003	MSVCR71.DLL
VC++ .NET 2005	MSVCR80.DLL
VC++ .NET 2008	MSVCR90.DLL

HBGary
DETECT. DIAGNOSE. RESPOND.

Static Linking

- C runtime library strings will be embedded in the EXE itself, as opposed to being in an external DLL
  - DOMAIN error
  - TLOSS error
  - SING error
  - R6027

HBGary
DETECT. DIAGNOSE. RESPOND.

Debug Symbols

- Debug timestamp (time_t – seconds since 01.01.1970)
- Version of the PDB file
  - NB09 - Codeview 4.10
  - NB11 - Codeview 5.0
  - NB10 - PDB 2.0
  - RSDS - PDB 7.0
- Age – number of times the malware has been compiled

HBGary
DETECT. DIAGNOSE. RESPOND.

Name Mangling

Compiler	void h(int)	void h(int, char)	void h(void)
Intel C++ 8.0 for Linux	_Z1hi	_Z1hic	_Z1hv
HP aC++ A.05.55 IA-64	_Z1hi	_Z1hic	_Z1hv
GNU GCC 3.*x* and 4.*x*	_Z1hi	_Z1hic	_Z1hv
HP aC++ A.03.45 PA-RISC	h__Fi	h__Fic	h__Fv
GNU GCC 2.9*x*	h__Fi	h__Fic	h__Fv
Microsoft VC++ v6/v7	?h@@YAXH@Z	?h@@YAXHD@Z	?h@@YAXXZ
Digital Mars C++	?h@@YAXH@Z	?h@@YAXHD@Z	?h@@YAXXZ
Borland C++ v3.1	@h$qi	@h$qizc	@h$qv
OpenVMS C++ V6.5 (ARM mode)	H__XI	H__XIC	H__XV
OpenVMS C++ V6.5 (ANSI mode)	CXX$__7H__FI0ARG51T	CXX$__7H__FIC26CDH77	CXX$__7H__FV2CB06E8
OpenVMS C++ X7.1 IA-64	CXX$_Z1HI2DSQ26A	CXX$_Z1HIC2NP3LI4	CXX$_Z1HV0BCA19V
SunPro CC	__1cBh6Fi_v_	__1cBh6Fic_v_	__1cBh6F_v_
Tru64 C++ V6.5 (ARM mode)	h__Xi	h__Xic	h__Xv
Tru64 C++ V6.5 (ANSI mode)	__7h__Fi	__7h__Fic	__7h__Fv
Watcom C++ 10.6	W?h$n(i)v	W?h$n(ia)v	W?h$n()v

HBGary
DETECT. DIAGNOSE. RESPOND.

Undecorate

Visual C++ demangle:

```
DWORD WINAPI UnDecorateSymbolName(
        __in PCTSTR DecoratedName,
        __out PTSTR UnDecoratedName,
        __in DWORD UndecoratedLength,
        __in DWORD Flags );
```

Also, see source to winedbg

GNU C++ demangle
see libiberty/cplus-dem.c and include/demangle.h

HBGary
DETECT. DIAGNOSE. RESPOND.

Delphi

- Give-away strings:

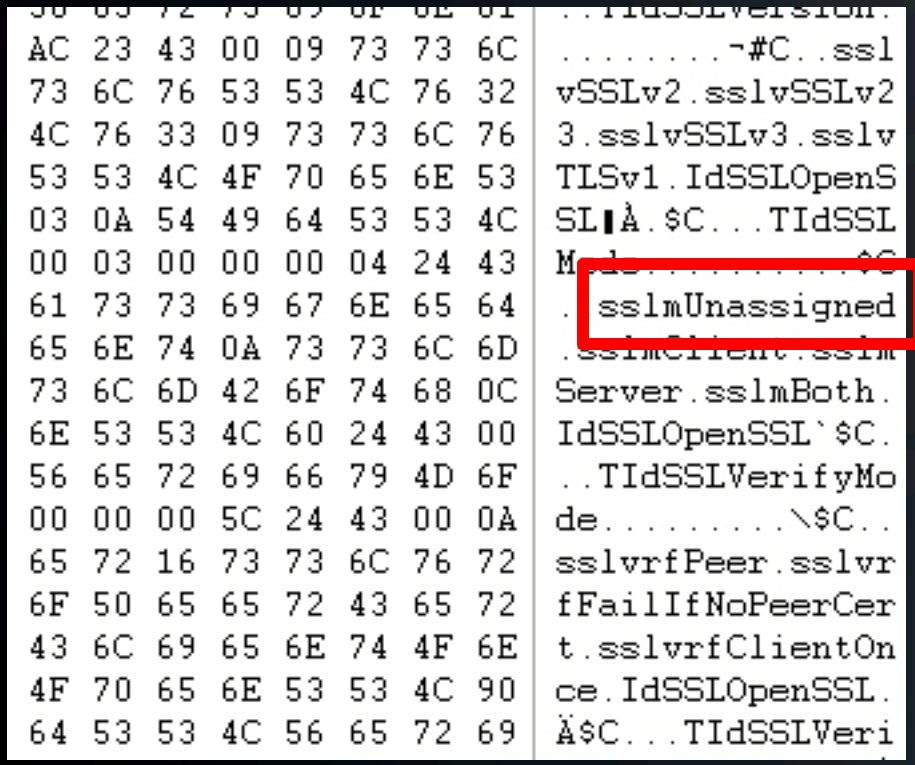
SOFTWARE\Borland\Delphi\RTL

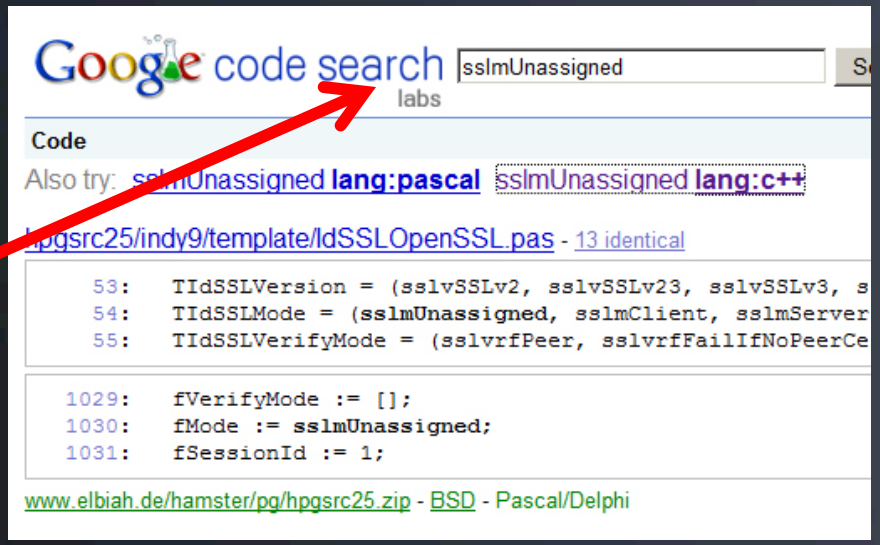
This program must be run under Win32

HBGary
DETECT. DIAGNOSE. RESPOND.

Delphi

- Uses specific function names – easy to identify
- Language is derived from Pascal





78 hits for pascal, only 2 for c++

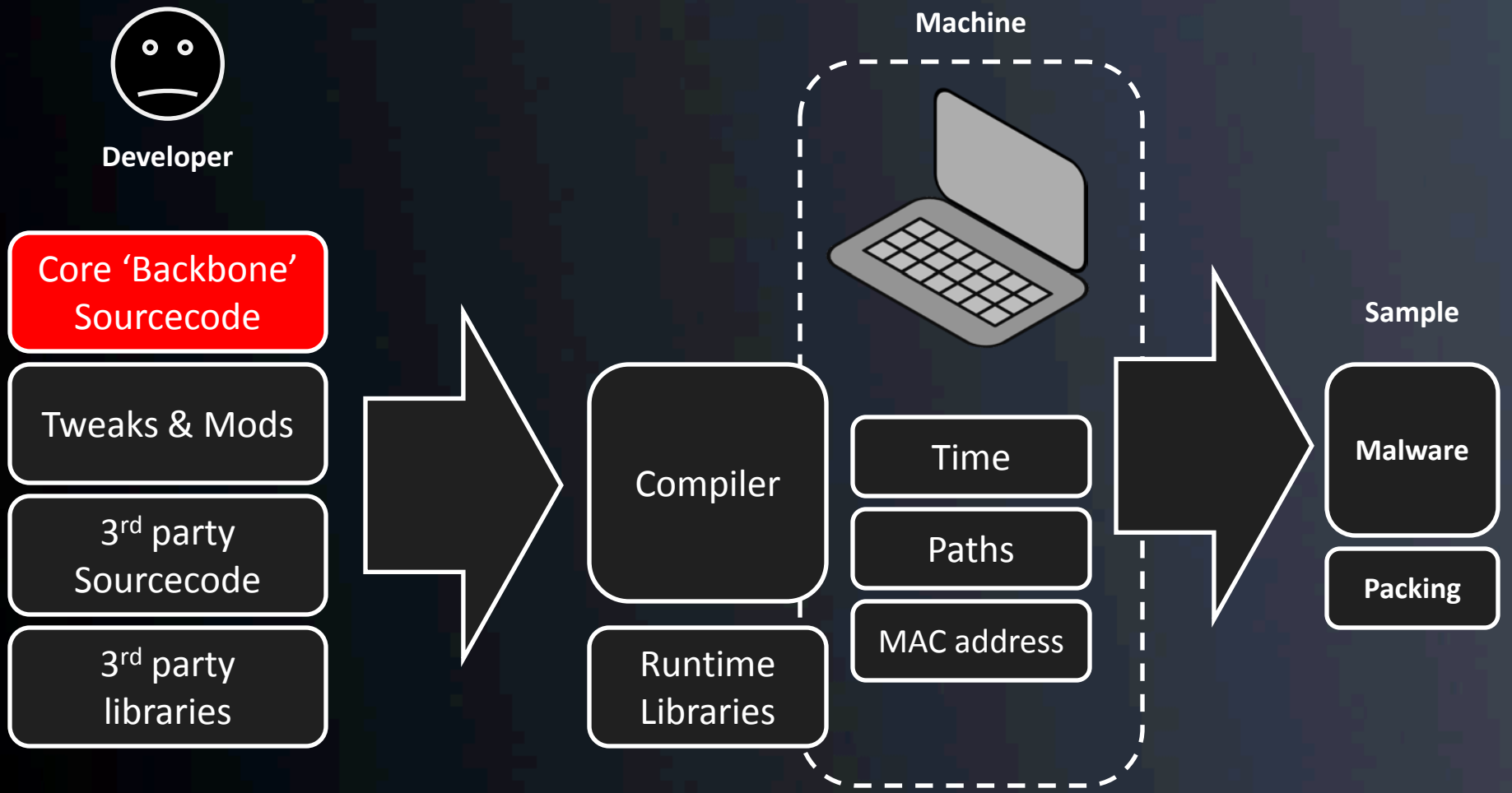
HBGary
DETECT. DIAGNOSE. RESPOND.

Embedded Manifest

- Contains name, description, platform
- Contains list of dependent modules + versions
  - May contain key tokens that identify specific dependent modules (aka strongly named)
- May contain public key that is tied to the developer if assembly itself is strongly named
  - not likely!
  - Public/private key pair (sn.exe)

HBGary
DETECT. DIAGNOSE. RESPOND.

Tracking Source Code



HBGary
DETECT. DIAGNOSE. RESPOND.

Main Functions

- Main
  - Same argument parsing
  - Init of global variables
  - WSAStartup
- DllMain
- ServiceMain

HBGary
DETECT. DIAGNOSE. RESPOND.

Service Routines

- Install / Uninstall Service
- RunDll32
- Service Start/Stop
- ServiceMain
- ControlService

Skeleton of a service

```
DllMain()
{
  // store the HANDLE to the module in a global variable
}

ServiceMain()
{
  // RegisterServiceCtrlHandler & store handle to service in global
variable
  // call SetServiceStatus, set PENDING, then RUNNING
  // call to main malware function(s)
}

ServiceCtrlHandler_Callback
{
  // handle various commands, start/stop/pause/etc
}
```

Size of local buffer

Sleep loop at end

dwWaitHint

Hard coded sleep( ) times

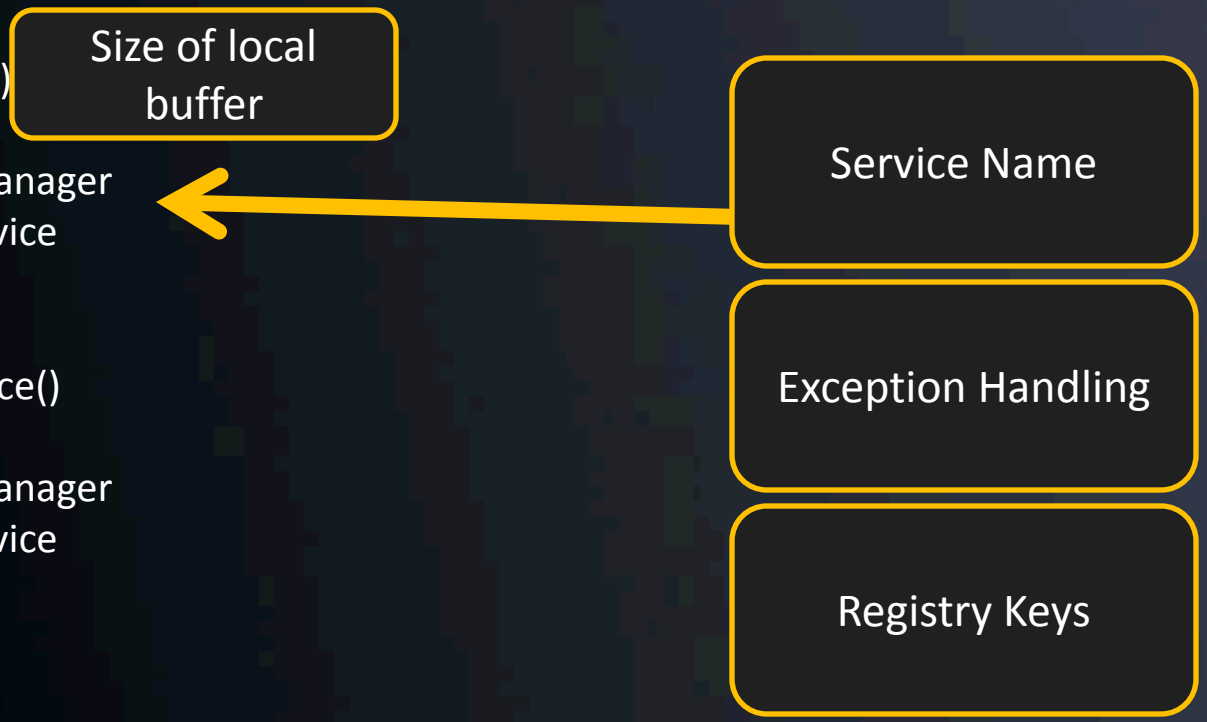
HBGary
DETECT. DIAGNOSE. RESPOND.

Skeleton of a service

```
Main_Malware_Function
{
  // do stuff
}

InstallService()
{
 // OpenSCManager
 // CreateService
}

UninstallService()
{
 // OpenSCManager
 // DeleteService
}
```



HBGary
DETECT. DIAGNOSE. RESPOND.

Filename Creation

- Log files, EXE's, DLL's
- Subdirectories
- Environment Variables
- Random numbers

Case Study: Chinese APT


```
65 45 78 28  RegQueryValueEx(
6E 74 65 72  Parameters\Inter
72 61 63 74  active).Interact
56 61 6C 75  ive.RegQueryValu
72 73 5C 70  eEx(Parameters\p
72 61 6D 00  rogram).program.
6E 74 43 6F  SYSTEM\CurrentCo
76 69 63 65  ntrolSet\Service
65 72 73 00  s\%s\Parameters
78 65 00 00  SvcHostDLL.exe..
0A 00 00 00  Sleep...Init....
66 69 6C 65  read remote file
66 69 6C 65   error!#....file
64 21 23 00   download end!#.
64 61 74 61  downend.downdata
25 64 00 00  ....datasize%d..
```

作 者	主题:SvcHostDll.dll
dargoner 化零为整 积分: 6 贴数: 5	日期 2005-3-10 8:35:50

```
#include <stdio.h>
#include <windows.h>
#include <time.h>

#define DEFAULT_SERVICE "IPRIP"
#define MY_EXECUTE_NAME "SvcHostDLL.exe"

//main service process function
void __stdcall ServiceMain( int argc, wchar_t* argv[] );
```

2005 posting of similar source code, includes poster's handle.

HBGary
DETECT. DIAGNOSE. RESPOND.

Case Study: Chinese APT


#define MY_EXECUTE_NAME "SvcHostDLL.exe" Search

About 426 results (0.56 seconds) Advanced search

Tip: Search for English results only. You can specify your search language in Preferences

svchostdll.rar svchostdll.cpp
... #define DEFAULT_SERVICE "IPRIP" #define MY_EXECUTE_NAME "SvcHostDLL.exe" DWORD ... see svchostdll.h for the class definition CSvchostdll::CSvchostdll() ...
read.pudn.com/downloads54/sourcecode/.../svchostdll.cpp__.htm - Cached

SvcHostDll.dll--补天论坛::补天网::Patching.net::0day-exploits::网 ...
Mar 10, 2005 ... #define DEFAULT_SERVICE "IPRIP" #define MY_EXECUTE_NAME "SvcHostDLL.exe" //main service process function void __stdcall ServiceMain(int ...
www.patching.net/bbs/viewdoc_43201_2.html - Cached - Similar

svchost难题，请高手请进- VC/MFC / 进程/线程/DLL - [Translate this page]
2006年7月12日 ... #define DEFAULT_SERVICE "IPRIP " #define MY_EXECUTE_NAME "SvcHostDll.exe " HANDLE hDll=NULL; SERVICE_STATUS_HANDLE hSrv; DWORD dwCurrState; ...
topic.csdn.net/t/20060712/01/4874487.html - China - Cached

svchost 服务怎么写? - [Translate this page]
8 posts - 5 authors - Last post: Jun 25, 2009
... #define DEFAULT_SERVICE "IPRIP" #define MY_EXECUTE_NAME "SvcHostDLL.exe" __declspec(dllexport) void __stdcall ServiceMain(int argc, ...
topic.csdn.net/.../5216321b-abe3-4197-bbf6-9417592b7e7c.html - China - Cached

Show more results from topic.csdn.net

XFOCUS Security Forums -> Re: bingle 请进，关于哪个svchost启动服务 ... - [Translate this page]
#define MY_EXECUTE_NAME "SvcHostDll.exe" HANDLE hDll=NULL; SERVICE_STATUS_HANDLE hSrv; DWORD dwCurrState; void _stdcall ServiceMain(int argc,wchar_t* ...
https://www.xfocus.org/bbs/index.php?act=SE&f=3&t=60693&p...

Continued searching will reveal many, many references to the base source code of this malware.

All malware samples for this attacker are derived from this basic framework, but many additions & modifications have been made.

HBGary
DETECT. DIAGNOSE. RESPOND.

3rd Party SourceCode

Developer

Core 'Backbone' Sourcecode

Tweaks & Mods

3rd party Sourcecode

3rd party libraries

Machine

Compiler

Runtime Libraries

Time

Paths

MAC address

Sample

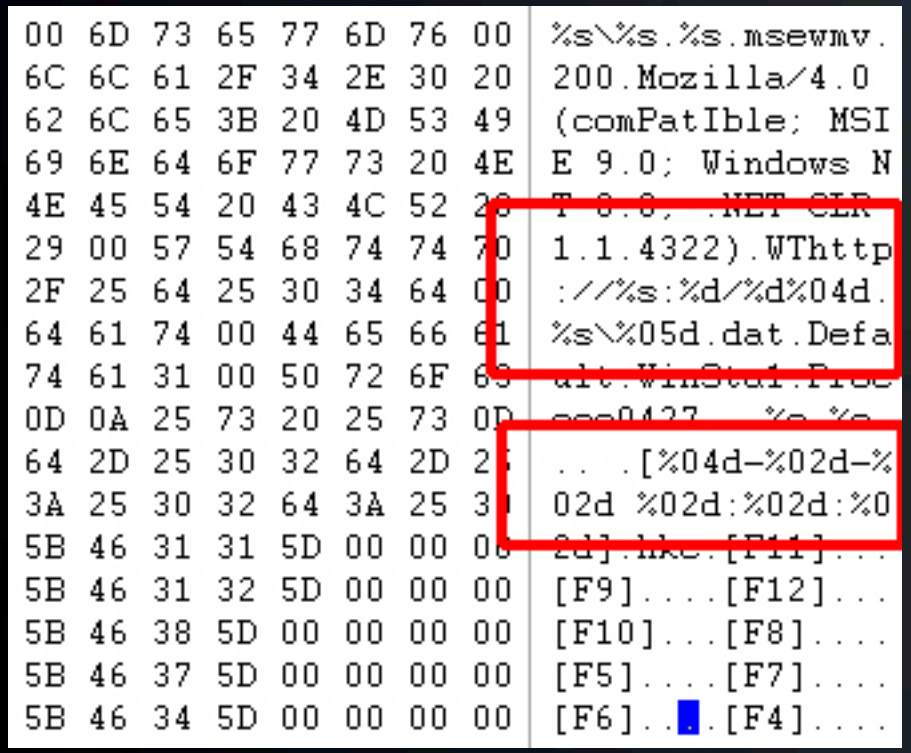
Malware

Packing

HBGary
DETECT. DIAGNOSE. RESPOND.

Format Strings

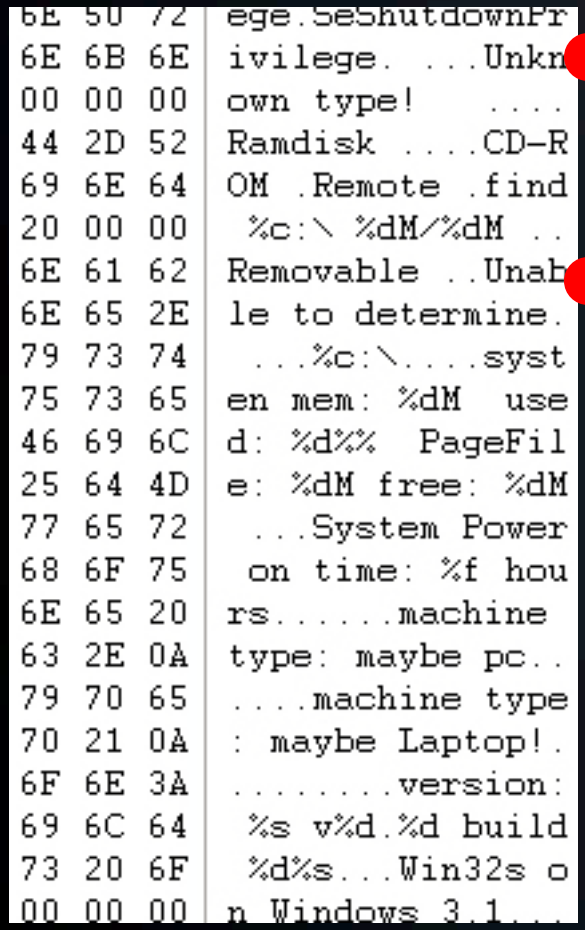
- These are written by humans, so they provide good uniqueness



http://%s:%d/%d%04d

HBGary
DETECT. DIAGNOSE. RESPOND.

Logging Strings



Searching for:
-“Unable to determine” &
-“Unknown type!”

Reveals that the attacker is using the source-code of BO2k for cut-and-paste material.

HBGary
DETECT. DIAGNOSE. RESPOND.

Google code search labs "Unable to determine" "Unknown type Search Advanced Code Search

Code

boxp_beta7/srv_system/main.h - 1 identical

```
81:   char    *sRplmeminfo;                 // Reply: "Memory: %dM in use: %d%%  Page file: %dM free: %dM\n"
82:   char    *sRplerrdsk;                  // Reply: "Unable to determine.\n"
83:   char    *sRpldskrmv;                  // Reply: "Removable\n"
```

```
87:   char    *sRpldskram;                  // Reply: "Ramdisk\n"
88:   char    *sRpldskuk;                           // Reply: "Unknown type!\n"
89:   char    *sRpldskinfo;                 // Reply: " Bytes free: %u MB(%s)/%u MB(%s)\n"
```

prdownloads.sourceforge.net/boxp/boxp_beta7_src.zip - GPL - C - More from boxp_beta7_src.zip »

boxp_beta6/srv_system/cmd_system.cpp - 1 identical

```
510:   case 0:
511:           api->plstrcat(svReply, "Unable to determine.\n");
512:           break;
```

```
548:   default:
549:           api->plstrcat(svReply, "Unknown type!\n");
550:           break;
```

prdownloads.sourceforge.net/boxp/boxp_beta6_src.zip - GPL - C++

srv_system/cmd_system.cpp - 2 identical

```
334:   case 0:
335:           lstrcat(svReply, "Unable to determine.\n");
336:           break;
```

```
360:   default:
361:           lstrcat(svReply, "Unknown type!\n");
362:           break;
```

prdownloads.sourceforge.net/bo2k/bo2kdev_src_1-1-1.zip - LGPL - C++

HBGary
DETECT. DIAGNOSE. RESPOND.

Mutex Names

```
73 5C 25 73 00 00 00 00  \Services\%s....
73 2E 25 73 00 00 00 00  rb..%s\%s.%s....
4C 41 59 00 44 65 66 61  tmp.DISPLAY.Defa
74 61 30 00 50 4F 53 54  ult.WinSta0.POST
00 00 00 00 4D 6F 7A 69  ....%d%s....Mozi
28 63 6F 6D 70 61 74 69  lla/4.0 (compati
45 20 36 2E 30 3B 20 57  ble; MSIE 6.0; W
54 20 35 2E 30 3B 20 2E  indows NT 5.0; .
31 2E 31 2E 34 33 32 34  NET CLR 1.1.4324
72 74 2E 75 69 64 00 66  ).w\somr.uid.f
00 00 20 00 68 6B 65 00  PsKey400.  .hke.
32 30 30 30 31 2E 74 6D  %d%d.f220001.tm
73 00 00 00 25 73 5C 73  p...%s\%s...%s\s
78 65 20 2D 6B 20 6E 65  vchost.exe -k ne
53 63 68 65 64 75 6C 65  tsvcs...Schedule
61 73 6
53 65 5
65 67 6
72 6F 7
75 72 7
6F 00 0
```

```
10006A1F   call_CreateMutexA:
10006A1F       mov eax,dword ptr [ebp+0x24]
10006A22       add esp,0x14
10006A25       shr eax,1
10006A27       push 0x100131F0:lpName_PsKey400
10006A2C       push 0x0:bIn
10006A2E       push 0x0:lpMutexAttributes
10006A30       mov ebx,0x1
10006A35       mov dword ptr [ebp+0x24],eax
10006A38       call dword ptr [0x100100D8] // __imp_KERNEL32.dll!CreateMutexA[000120D6]
```

Mutex names remain consistent at least for one infection-push, as they are designed to prevent multiple-infections for the same malware.

HBGary
DETECT. DIAGNOSE. RESPOND.

Link Analysis

Hook键盘记录器的问题。。。。。

今天搞了一下Hook键盘记录器。。。。。

不知道为会么写文件的时候会出错。。

贴关键代码。。。。看来得解决这个问题才行啊。。。。。。。。。

```
void WriteChar(char* sText)
{
 //加锁
 HANDLE hMetux = OpenMutex(MUTEX_ALL_ACCESS, FALSE, "PsKey400");
 if(hMetux != NULL)
  WaitForSingleObject(hMetux, 300);

 FILE fp;
 if ((fp = &fopen(m_CharFileName,"ab")) == NULL)
 {
  MessageBox(NULL,"打开了出错","打开了出错",MB_OK);
  fclose(&fp);
 }
 if (fwrite(sText,strlen(sText),1,&fp) != 1)
 {
  MessageBox(NULL,"写入出错","写入出错",MB_OK);
  fclose(&fp);
 }
 fclose(&fp);
```

HBGary
DETECT. DIAGNOSE. RESPOND.

3rd Party Libraries

Developer

Core 'Backbone' Sourcecode

Tweaks & Mods

3rd party Sourcecode

3rd party libraries

Machine

Compiler

Runtime Libraries

Time

Paths

MAC address

Sample

Malware

Packing

HBGary
DETECT. DIAGNOSE. RESPOND.

Copyright & Version Strings

OpenSSL/0.9.6
RAND part of OpenSSL 0.9.8e 23 Feb 2007
MD5 part of OpenSSL 0.9.8k 25 Mar 2009
libdes part of OpenSSL 0.9.7b 10 Apr 2003
inflate 1.2.1 Copyright 1995-2003 Mark Adler
inflate 1.1.4 Copyright 1995-2002 Mark Adler
inflate 1.2.3 Copyright 1995-2005 Mark Adler
inflate 1.0.4 Copyright 1995-1996 Mark Adler
inflate 1.1.3 Copyright 1995-1998 Mark Adler
inflate 1.1.2 Copyright 1995-1998 Mark Adler
inflate 1.2.2 Copyright 1995-2004 Mark Adler

HBGary
DETECT. DIAGNOSE. RESPOND.

zlib Fingerprinting

- Every new version of zlib has a unique pattern of bits in the data tables – these are modified for each version specifically
- This pattern is a data constant and can be used even if the copyright notices have been removed

http://www.enyo.de/fw/security/zlib-fingerprint/zlib.db

HBGary
DETECT. DIAGNOSE. RESPOND.

inflate library patterns

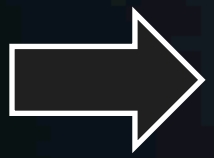
- Not as specific as zlib patterns but can be used to detect the inflate decompressor

http://www.enyo.de/fw/security/zlib-fingerprint/inflate.db

HBGary
DETECT. DIAGNOSE. RESPOND.

Installation & Deployment

Developer



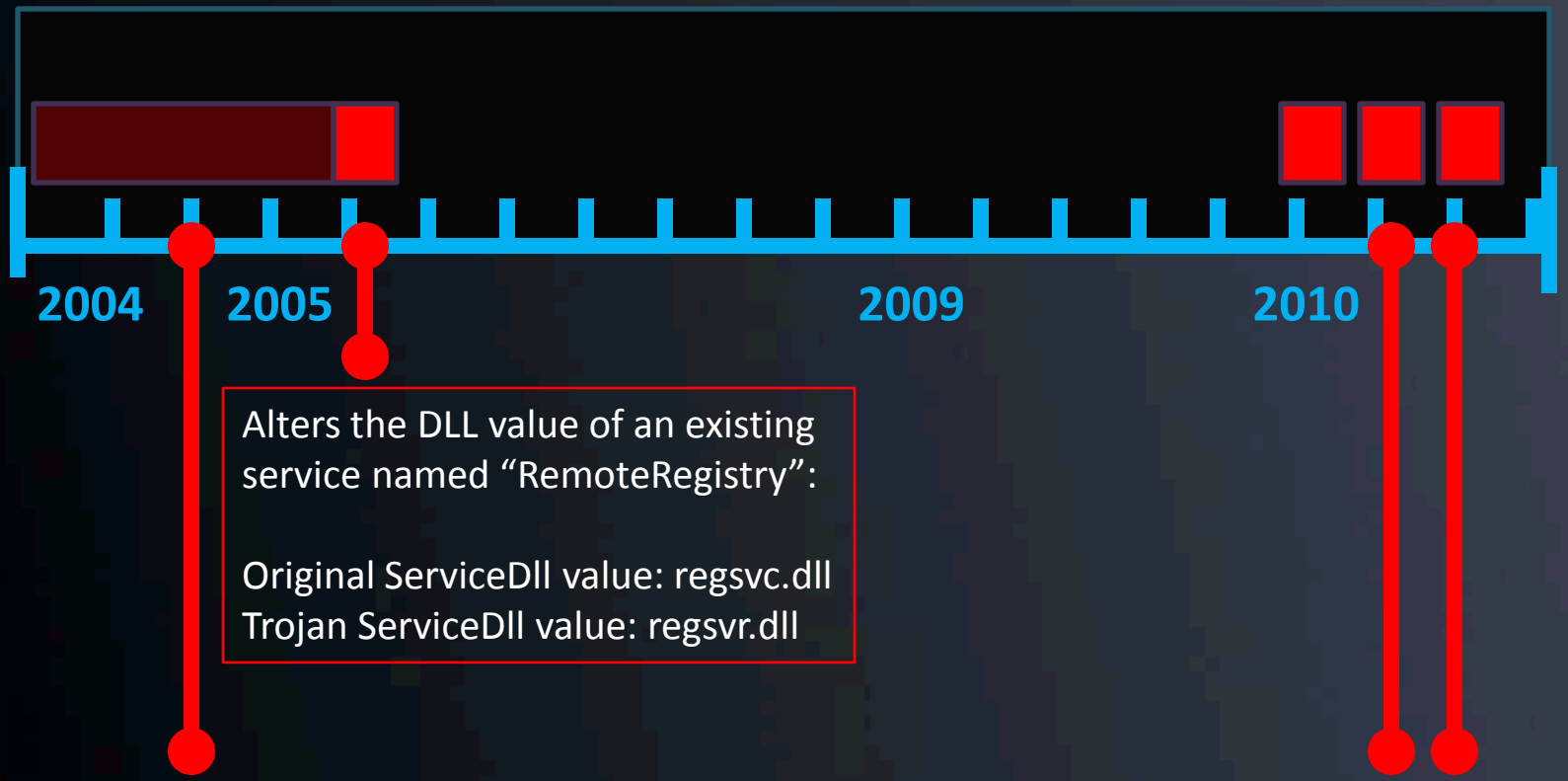
- Communications Functions
- **Installation & Deployment Method**
- Command & Control Functions
- Compiler Environment
- Stealth & Antiforensic Techniques

Sample

- Malware
- Packing

HBGary
DETECT. DIAGNOSE. RESPOND.

Case Study: Chinese APT

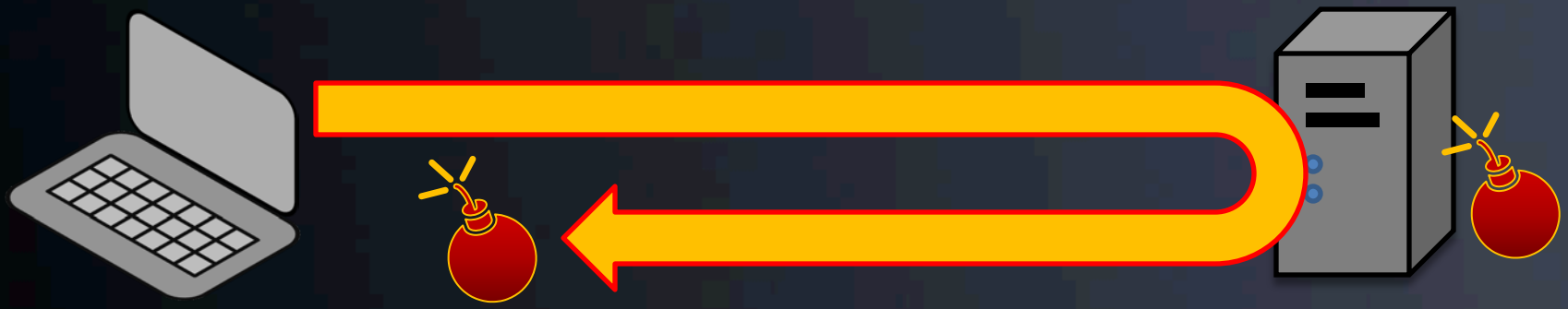


HBGary
DETECT. DIAGNOSE. RESPOND.

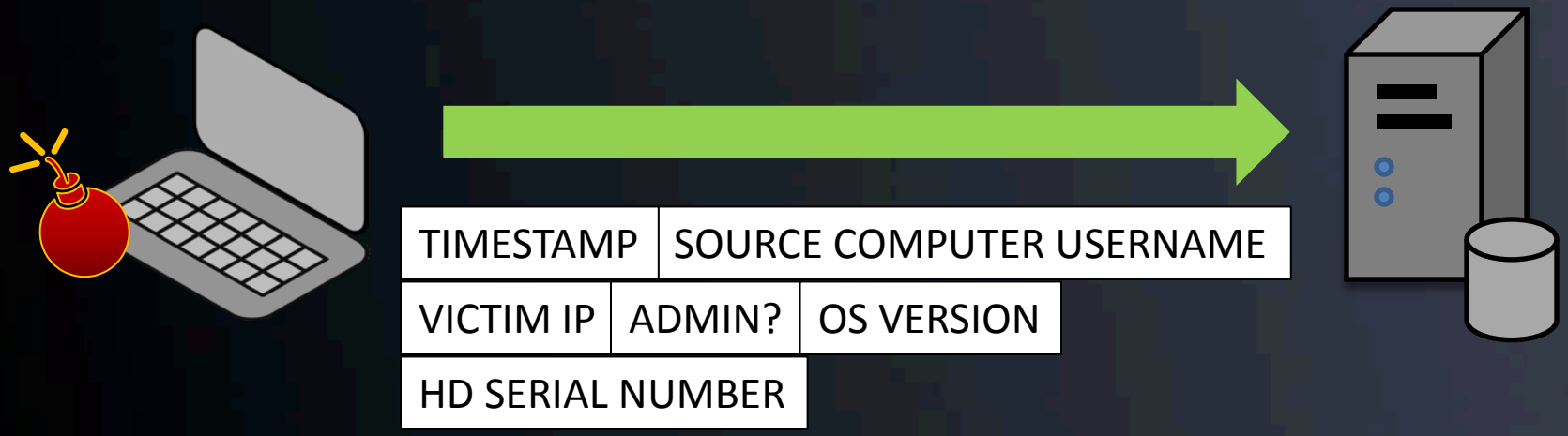
Command & Control

HBGary
DETECT. DIAGNOSE. RESPOND.

Command and Control

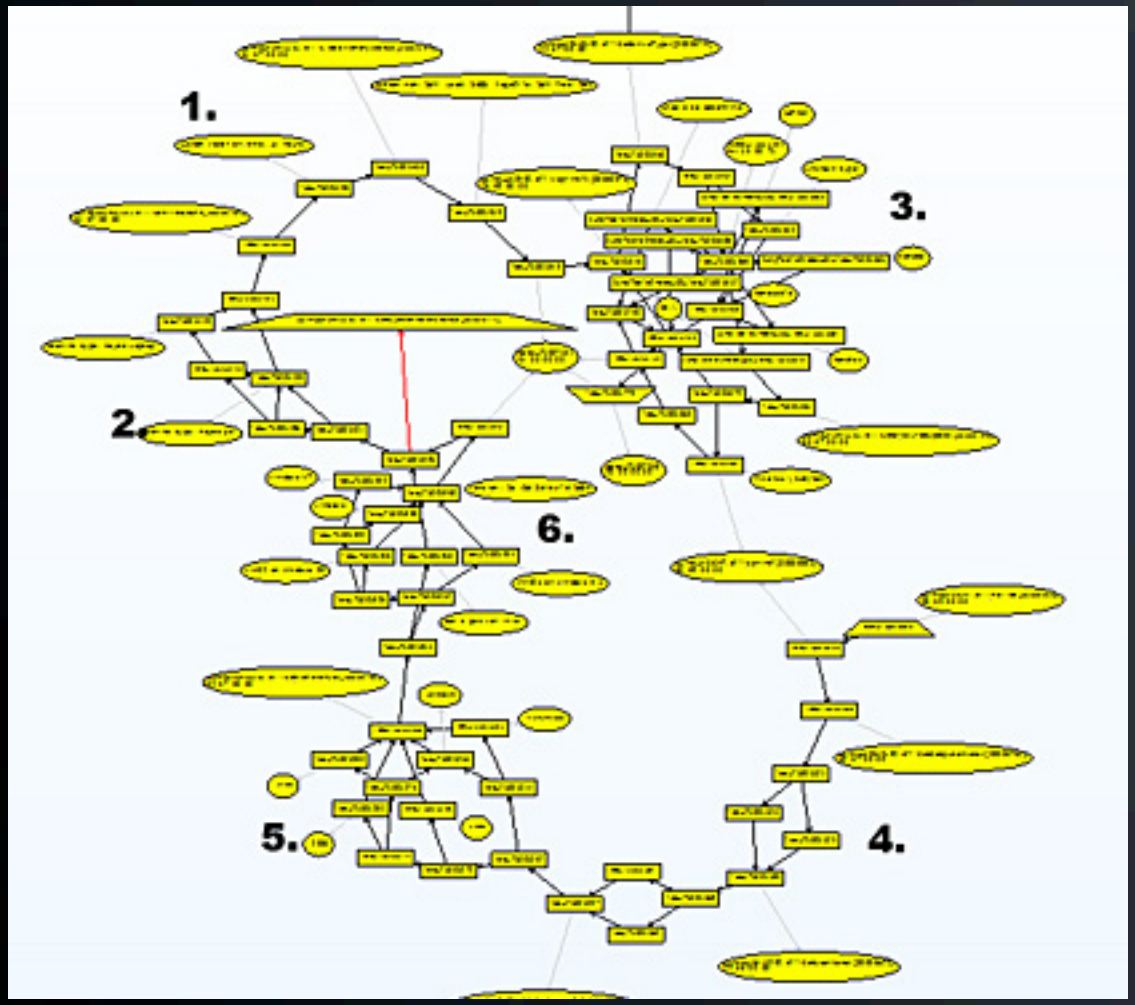


Once installed, the malware phones home…



HBGary
DETECT. DIAGNOSE. RESPOND.

C&C Hello Message



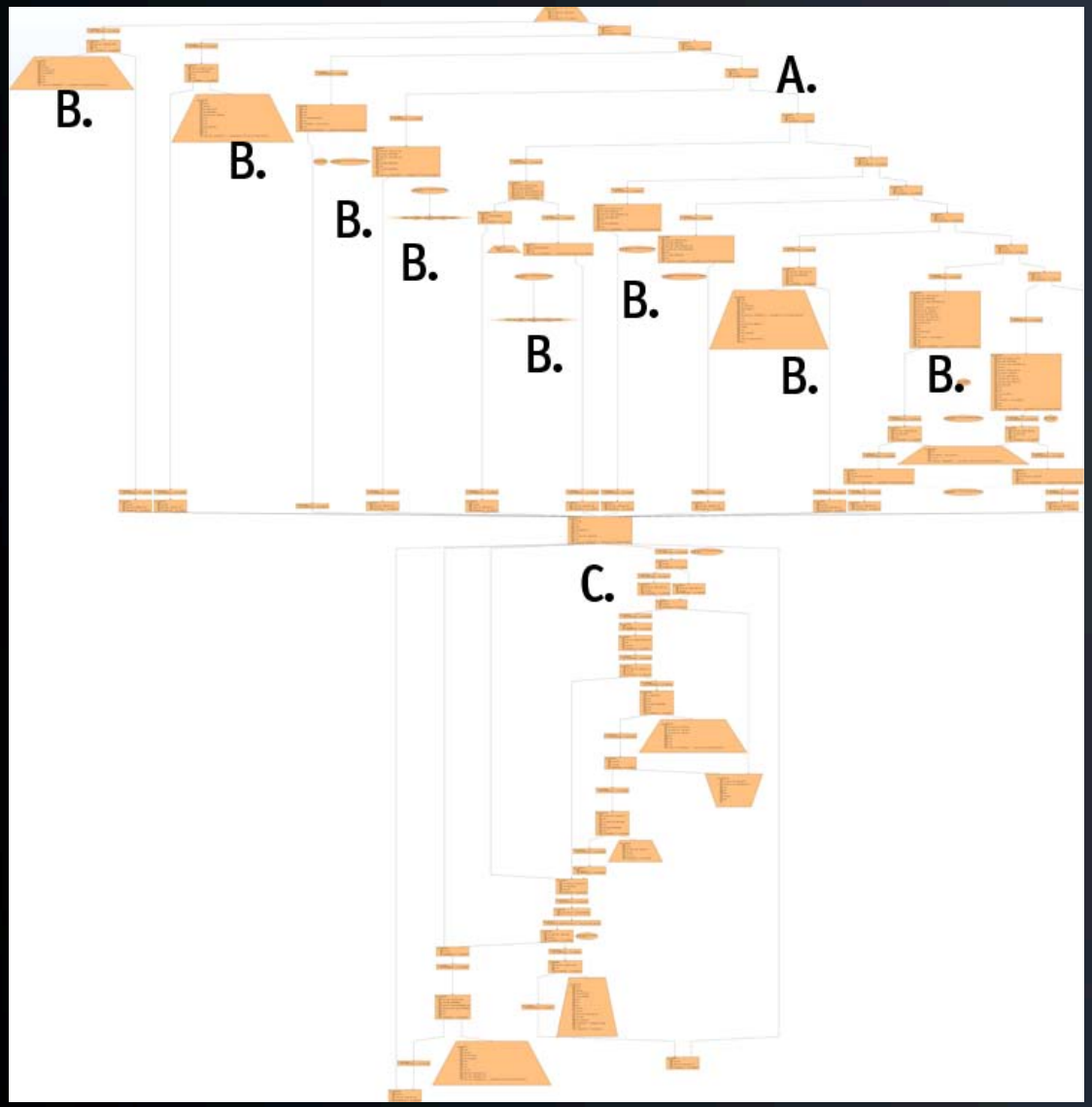
1) this queries the uptime of the machine..
2) checks whether it's a laptop or desktop machine...
3) enumerates all the drives attached to the system, including USB and network...
4) gets the windows username and computername...
5) gets the CPU info... and finally,
6) the version and build number of windows.

HBGary
DETECT. DIAGNOSE. RESPOND.

Command and Control Server

- The C&C system may vary
  - Custom protocol (Aurora-like)
  - Plain Old URL's
  - IRC (not so common anymore)
  - Stealth / embedded in legitimate traffic
- Machine identification
  - Stored infections in a back end SQL database

Aurora C&C parser



A) Command is stored as a number, not text. It is checked here.
B) Each individual command handler is clearly visible below the numerical check
C) After the command handler processes the command, the result is sent back to the C&C server

HBGary
DETECT. DIAGNOSE. RESPOND.

Advanced Fingerprinting

GhostNet: Screen Capture Algorithm

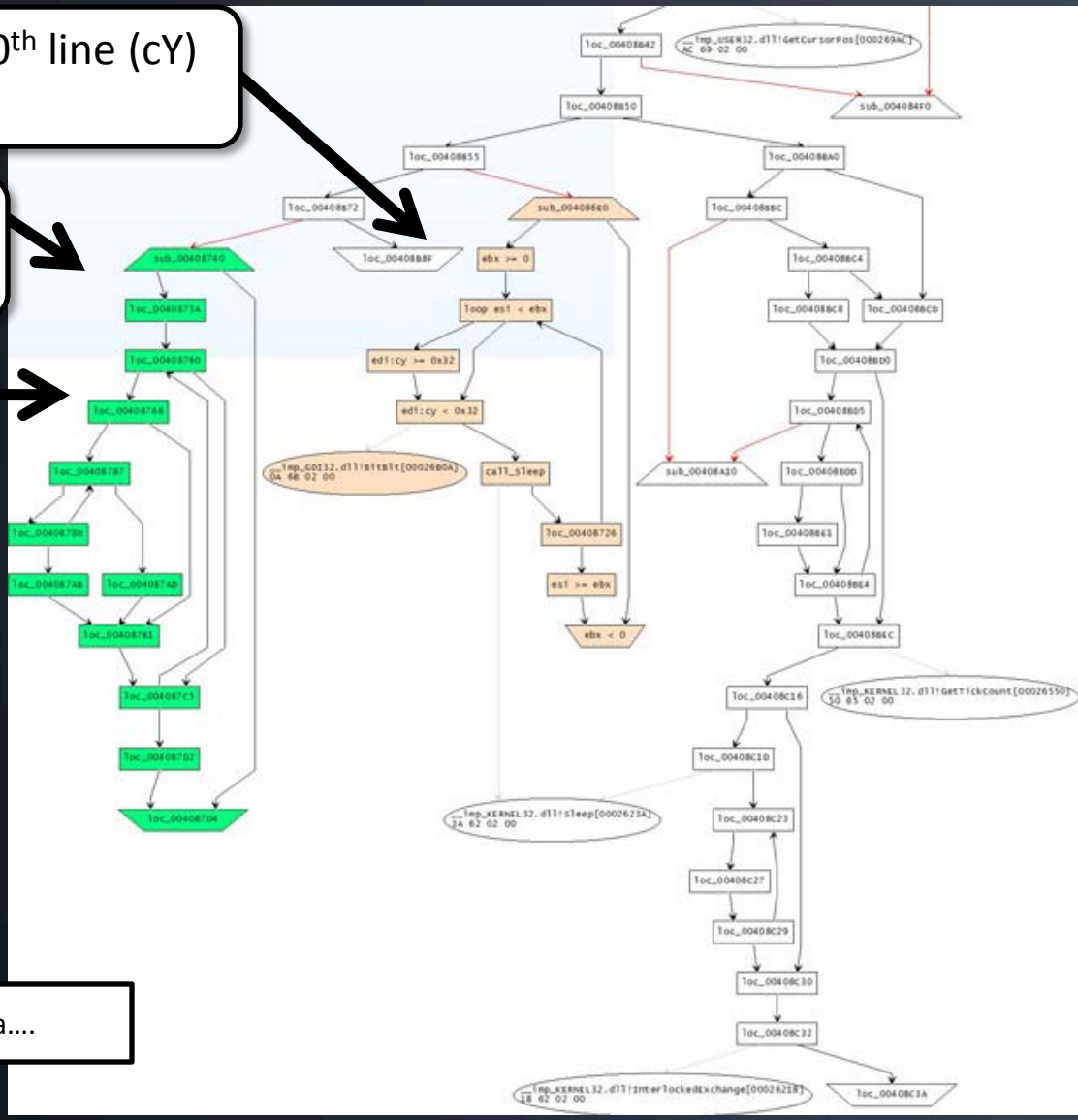
Loops, scanning every 50th line (cY) of the display.

Reads screenshot data, creates a special DIFF buffer

LOOP: Compare new screenshot to previous, 4 bytes at a time

If they differ, enter secondary loop here, writing a 'data run' for as long as there is no match.

Offset in screenshot	Len in bytes	Data….



GhostNet: Searching for sourcecode

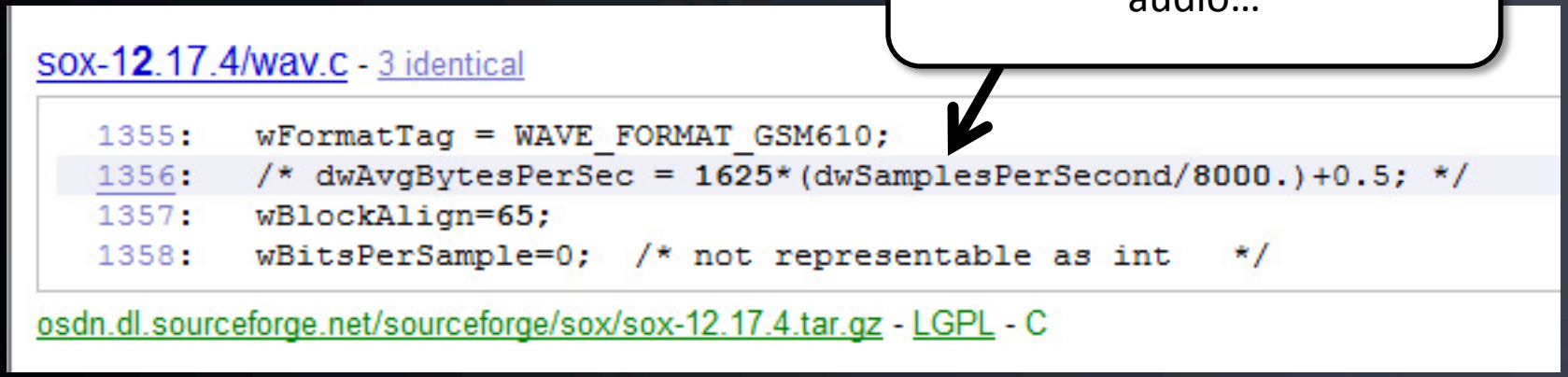
```
00401080        mov dword ptr [esi+0x56],eax
00401083        mov eax,0x1
00401088        mov edx,0x31
0040108D        mov word ptr [esi+0x48],ax
00401091        mov ecx,0x41
00401096        mov word ptr [esi+0x46],dx
0040109A        mov word ptr [esi+0x52],cx
0040109E        mov eax,0x2
004010A3        pop edi
004010A4        xor edx,edx
004010A6        mov word ptr [esi+0x56],ax
004010AA        mov ecx,0x0140
004010AF        mov dword ptr [esi+0x4A],0x1F40
004010B6        mov dword ptr [esi+0x4E],0x659
004010BD        mov word ptr [esi+0x54],dx
004010C1        mov word ptr [esi+0x58],cx
004010C5        mov eax,esi
004010C7        pop esi
004010C8        pop ebp
004010C9        pop ebx
004010CA        ret
```

Large grouping of constants

Search source code of the 'Net

GhostNet: Refining Search

Has something to do with audio…



Further refine the search by including 'WAVE_FORMAT_GSM610' in the search requirements…

GhostNet: Source Discovery

```
CAudio::CAudio()
{
        m_hEventWaveIn          = CreateEvent(NULL, false, false, NULL);
        m_hStartRecord          = CreateEvent(NULL, false, false, NULL);
        m_hThreadCallBack       = NULL;
        m_nWaveInIndex          = 0;
        m_nWaveOutIndex         = 0;
        m_nBufferLength         = 1000; // m_GSMWavefmt.wfx.nSamplesPerSec / 8(bit)

        m_bIsWaveInUsed         = false;
        m_bIsWaveOutUsed        = false;

        for (int i = 0; i < 2; i++)
        {
                m_lpInAudioData[i] = new BYTE[m_nBu
                m_lpInAudioHdr[i] = new WAVEHDR;

                m_lpOutAudioData[i] = new BYTE[m_nB
                m_lpOutAudioHdr[i] = new WAVEHDR;
        }

        memset(&m_GSMWavefmt, 0, sizeof(GSM610WAVEF

        m_GSMWavefmt.wfx.wFormatTag = WAVE_FORMAT_G
        m_GSMWavefmt.wfx.nChannels = 1;
        m_GSMWavefmt.wfx.nSamplesPerSec = 8000;
        m_GSMWavefmt.wfx.nAvgBytesPerSec = 1625;
        m_GSMWavefmt.wfx.nBlockAlign = 65;
        m_GSMWavefmt.wfx.wBitsPerSample = 0;
        m_GSMWavefmt.wfx.cbSize = 2;
```

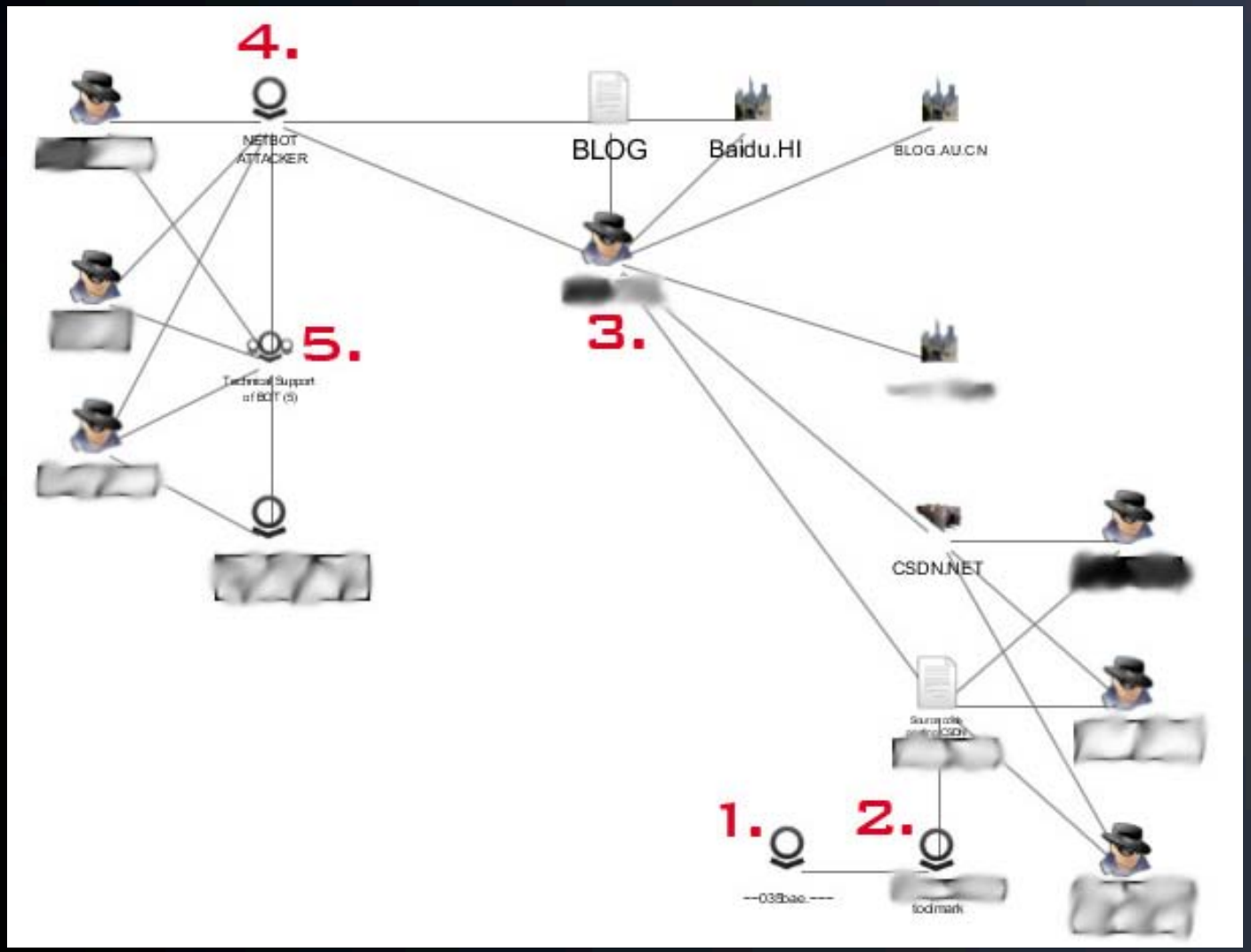
We discover a nearly perfect ‘c’ representation of the disassembled function. Clearly cut-and-paste.

We can assume most of the audio functions are this implementation of ‘CAudio’ class – no need for any further low-level RE work.

HBGary
DETECT. DIAGNOSE. RESPOND.

On link analysis…

Example: Link Analysis with Palantir™



1. Implant
2. Forensic Toolmark specific to Implant
3. Searching the 'Net reveals source code that leads to Actor
4. Actor is supplying a backdoor
5. Group of people asking for technical support on their copies of the backdoor

HBGary
DETECT. DIAGNOSE. RESPOND.

Working back the timeline

- Who sells it, when did that capability first emerge?
  - Requires ongoing monitoring of all open-source intelligence, presence within underground marketplaces
  - Requires budget for acquisition of emerging malware products

HBGary
DETECT. DIAGNOSE. RESPOND.

Conclusion

HBGary
DETECT. DIAGNOSE. RESPOND.

Takeaways

- Actionable intelligence can be obtained from malware infections ***for immediate defense:***
  - File, Registry, and IP/URL information
- Existing security doesn't stop 'bad guys'
  - Go 'beyond the checkbox'
- Adversaries have intent and funding
  - Failure is hiccup – doesn't stop mission
- Need to focus on the criminal, not malware
  - Attribution is possible thru forensic toolmarking combined with open and closed source intelligence

HBGary
DETECT. DIAGNOSE. RESPOND.

Continued Work

- Will be presenting additional research at BlackHat Vegas this year
  - Trend over 500k malware samples
- HBGary will be releasing a free tool that will dump fingerprint information from a binary or livebin

HBGary
DETECT. DIAGNOSE. RESPOND.

Fingerprint Utility

```
Developer Fingerprint Utility, Copyright 2010 HBGary, INC
File: 1228ad2e39befa4319733e98d8ed2890.livebin

Original project name:          RESSDT
Developer's project directory: e:\gh0st\server\sys\i386
Compiler:                       Microsoft Visual C++ 6.0 release

User interface:                 Windows GDI/Common Controls
Media:                          Windows multimedia API
Media:                          Microsoft VfW (Video for Windows)
Compression:                    Inflate Library version: 1.1.4
Networking:                     Windows sockets (TCP/IP)
Networking:                     Windows Internet API

Source directory:               e:\gh0st\server\sys\i386
```

HBGary
DETECT. DIAGNOSE. RESPOND.

Thank You

- HBGary, Inc. (www.hbgary.com)
- HBGary Federal (www.hbgaryfederal.com)