This little paper want's to inform you about the potential risk a BotNet or similar malicious software could cause to your system and your privacy. After a talk with a good friend, when we were talking about malware, i decided to give it a try and write a basic BotNet to show nearly everyone could create such a thing.

You need no supernatural power to build a Bot, just some basic Coding-KungFu ;)
You might ask, why someone would write a Bot and a Control-GUI, if he does not plan to use it?
The answer is simple: I wanted to know how far i would get with this project. How effective would it be and what problems would appear upon coding it? Sometimes you need to create it yourself to understand how things work! That is all. You can learn how to protect yourself and others by understanding such things in a deeper detail.

So the following is for educational purposes only. And i know the bad guys can do it much better. But for us this is a little overview, what is possible to do by just one person coding stuff from scratch to put you and your privacy at high risk. I am not a BlackHat nor do i call myself a hacker. I am just a guy that loves to code stuff and help people to keep their machines safe. This little paper is not to educate hackers or script kiddies. I do not support any illegal actions against anybody over the internet or in real life. I am not responsible for what you do with my given information…

Not all people do really care about their cyber-security – this has to change!

This is a call to „wake up" and finally care about it.

This Bot will further be called „GovernBot" and is an allusion to PRSIM and Tempora ;-)

So let us define what this little Bot can do with its functions (incomplete summary):

Functions:

Simple basic functions with no special input:

- Turn Webcam LED ON or OFF
- Start a Keylogger
- Block all Input to the machine
- Delete Eventlogs
- Open/ close CD-Drive
- Scan for available drives
- List all running process
- Start a DDoS Attack
- Gather Systeminformation (MACs,Internet-IP,LAN-IP,PCName etc.)
- Force Reboot, Logoff or Shutdown
- Update itself & delete itself without leaving a trace
- Keylogging

Advanced functions, that need more input:

- Capture pictures via your webcam (enable/ disable LED)
- Take a Screenshot from your monitor
- Record Audiostreams
- Sent SMTP-Mails (fully faked/ spoofed) with attachments
- Start a Program or URL
- Force a http(s)-Download (silent)
- Upload singel files via FTP
- Upload complete folders via FTP
- DDoS (basic slowloris, http-Flooding, Ping of Death)
- Timed Shutdown
- Un-/Zip files & folders, optional with Password
- Process-Viewer (can terminate a process)
- Popup Fakemessage
- Play Sounds
- Scan available WLAN's nearby
- Browse files on the infected machine and download or upload stuff
- SSH Remote-Tunnel
  (Attacker <<->> http-server (sending commands) <<->>
  Bot/Client (forward commands) <<->> SSH-Server (executing commands and send command-prompt back all the way)

Built with and controled via?

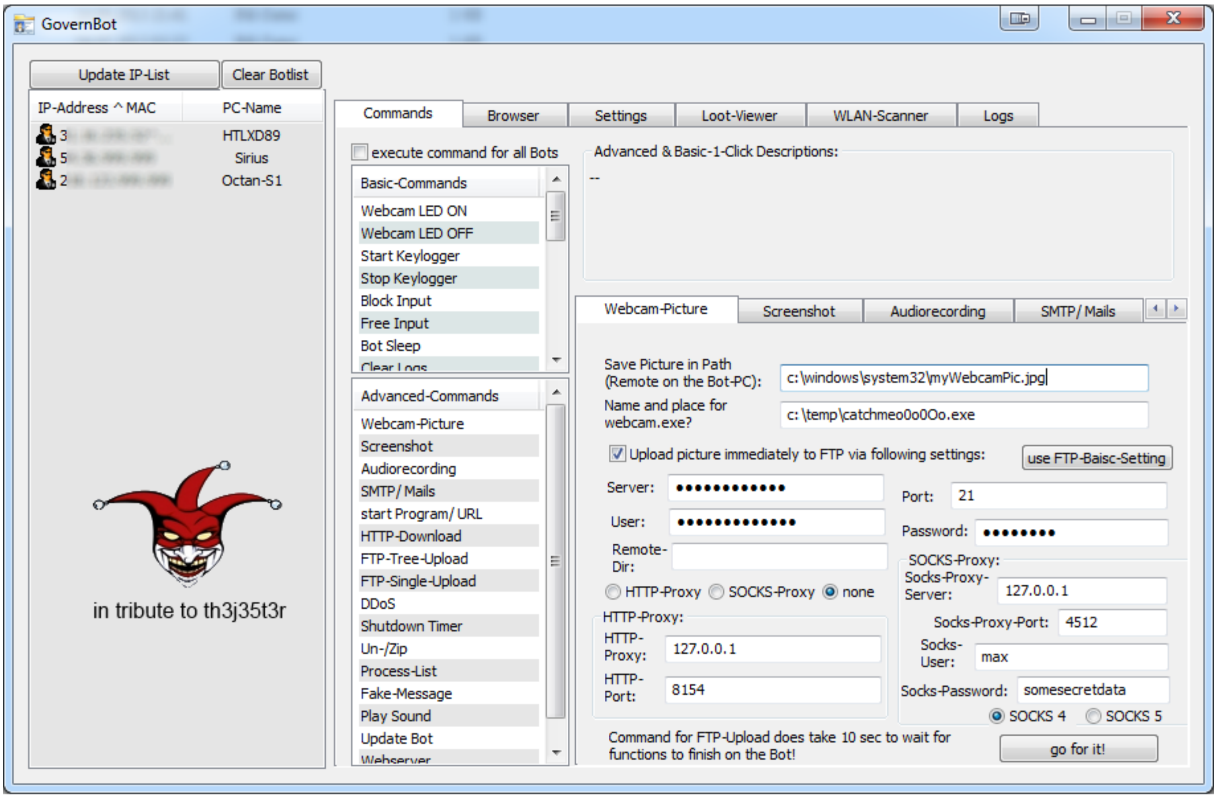
As you can see there are a few nice things you can do with GovernBot. And the most important stuff is, that most AV and Firewall-Solutions i tested won't stop this Bot from interacting with it's control server! The reason is, it uses common standard ports and protocols which are often not monitored enough.

The Bot is a simple EXE-File (written in C/C++ and MFC/WinAPI) which, after it was started, will try to connect to a http-Server every 5 seconds, if it finds an active connection to the Internet. Therefor it will do a basic authentication (keyword: .htaccess / .htpasswd) on the http-Server to access some underlying .php-Files that will filter and store all the requests from the Bot. The basic-auth is used to prevent other visitors to see what happens on our http-Server (well yes, you could sniff it i know…)

All http-requests from and to the Bot are encrypted with a special self developed algo, that is able to transport our POST or GET in a more secure way to the server.

To control our Bots we don't use a webinterface. Instead i have written a GUI to manage all the tasks. Pro: nobody can hijack your bots by simply cracking the webinterface.

The GUI (graphical user interface) used by the attacker looks like this:



IP-Address ^ MAC	PC-Name
3 [redacted]	HTLXD89
5 [redacted]	Sirius
2 [redacted]	Octan-S1

□ execute command for all Bots

Advanced & Basic-1-Click Descriptions:
--

Save Picture in Path (Remote on the Bot-PC): c:\windows\system32\myWebcamPic.jpg
Name and place for webcam.exe? c:\temp\catchmeo0o0Oo.exe
☑ Upload picture immediately to FTP via following settings:
Server: ●●●●●●●●●●●●
Port: 21
User: ●●●●●●●●●●●●●
Password: ●●●●●●●●
Remote-Dir:
○ HTTP-Proxy ○ SOCKS-Proxy ● none
HTTP-Proxy:
HTTP-Proxy: 127.0.0.1
HTTP-Port: 8154
SOCKS-Proxy:
Socks-Proxy-Server: 127.0.0.1
Socks-Proxy-Port: 4512
Socks-User: max
Socks-Password: somesecretdata
● SOCKS 4 ○ SOCKS 5
Command for FTP-Upload does take 10 sec to wait for functions to finish on the Bot!

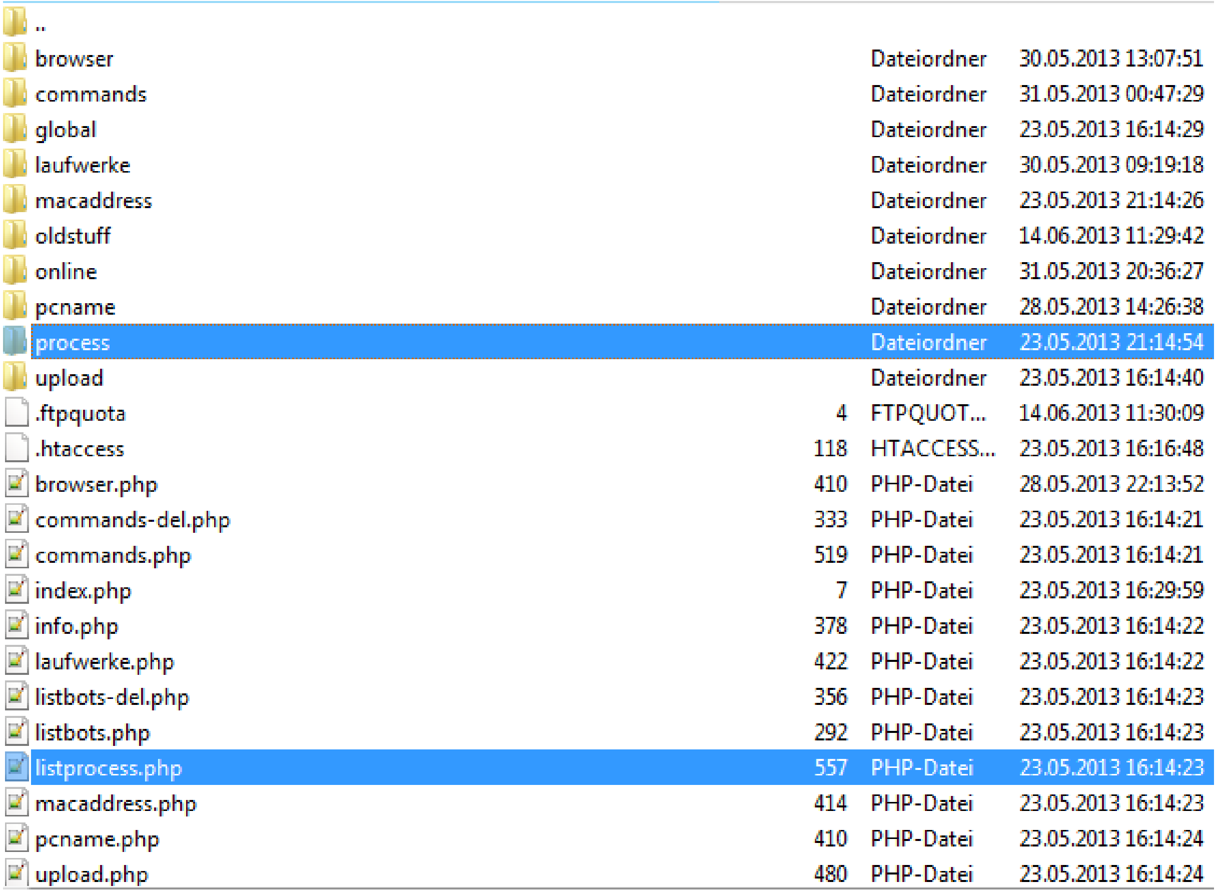
Motivated and inspired by @th3j35t3r, i did add a little tribute to him, with his friendly permission ;-P And this does NOT mean i know him, or he would use any kind of bots, or is affiliated to this little paper about GovernBot in any way… etc. etc. – so stay calm and frosty ^^

Let us have a quick look at the structure and a function (e.g. to list all running processes). That function would consist of e.g. this partial-code (nearly useless partial code in this paper only, to prevent abuse):

```
//……
HANDLE proc;
int i = 0;
PROCESSENTRY32 process;
void* photo = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);

Process32First(photo,&process);
//……
```

After it got all the running process it would try to put it all together into a string (separated by </br>) and transmit it to our server (encrypted) with a synchronous GET-request. We have to take care that our string within that URL does not get too long, because there is a limit of chars you are able to submit (optional way is, to transfer it via FTP).

The file accepting our request is stored on the servers root, but does save the received content in a subfolder (/process). I wanted to keep things simple so here is the structure on the server in a screenshot:



..			
browser		Dateiordner	30.05.2013 13:07:51
commands		Dateiordner	31.05.2013 00:47:29
global		Dateiordner	23.05.2013 16:14:29
laufwerke		Dateiordner	30.05.2013 09:19:18
macaddress		Dateiordner	23.05.2013 21:14:26
oldstuff		Dateiordner	14.06.2013 11:29:42
online		Dateiordner	31.05.2013 20:36:27
pcname		Dateiordner	28.05.2013 14:26:38
process		Dateiordner	23.05.2013 21:14:54
upload		Dateiordner	23.05.2013 16:14:40
.ftpquota	4	FTPQUOT...	14.06.2013 11:30:09
.htaccess	118	HTACCESS...	23.05.2013 16:16:48
browser.php	410	PHP-Datei	28.05.2013 22:13:52
commands-del.php	333	PHP-Datei	23.05.2013 16:14:21
commands.php	519	PHP-Datei	23.05.2013 16:14:21
index.php	7	PHP-Datei	23.05.2013 16:29:59
info.php	378	PHP-Datei	23.05.2013 16:14:22
laufwerke.php	422	PHP-Datei	23.05.2013 16:14:22
listbots-del.php	356	PHP-Datei	23.05.2013 16:14:23
listbots.php	292	PHP-Datei	23.05.2013 16:14:23
listprocess.php	557	PHP-Datei	23.05.2013 16:14:23
macaddress.php	414	PHP-Datei	23.05.2013 16:14:23
pcname.php	410	PHP-Datei	23.05.2013 16:14:24
upload.php	480	PHP-Datei	23.05.2013 16:14:24

Unencrypted the file within the process-folder looks like this:

```
Anzahl=30</br>Prozesse=</br>System</br>smss.exe</br>csrss.exe</br>wininit.exe</br>services.ex
e</br>lsass.exe</br>lsm.exe</br>svchost.exe</br>svchost.exe</br>crazytunes.exe</br>svchost.exe
</br>svchost.exe</br>svchost.exe</br>svchost.exe</br>RootSrv.exe</br>CTAudSvc.exe</br>svchos
t.exe</br>spoolsv.exe</br>svchost.exe</br>svchost.exe</br>armsvc.exe</br>adpsrvs.exe</br>Fuel.
Service.exe</br>monder.exe</br>ceps.exe</br>monsec.exe</br>idchkbka.exe</br>
```

If our GUI picks up the stored data later we can simply parse the </br> and handle each process as one line for a ListControl etc. for easy editing and action-handling like „terminate process“.

If you have to manage a lot of different machines you may want not only to identify each machine later again, you would like to infect other machines within the same LAN later if you managed your way into a network, and control them as well (GovernBot has some LAN-Enumeration and scanning engines on board).

This is why GovernBot does grab all MAC-Address and does create a unique string like 35.154.47.41^00-08-65-d1-01-02-Username. The first part is the Bot-Internet-IP. To get and verify this one you let the Bot check against different sites who will be parsed for the detected IP-Address. That is how you can catch different machines on the same public Internet-IP.

The second part after the „^“ is the first found MAC-Address on the machine.
Last value is the currently logged in User(name) on the machine (remember, this could

be a companies server with multiple user-sessions open and we might want to target a specific individual working at this company ;-)

As you can see in the Screenshot for the GUI the Machine-Name is shown as well. So you should be able to find a machine pretty fast if it is back online.

The lost privacy!

Maybe you did read about the CERT-Georgia, that catched a hacker with his own malware on his webcam? If not you can read it here: **http://tinyurl.com/kp4uaep**
I thougt it would be a nice thing do add a webcam function as well for demonstration ;-P

In a separated project i did create my webcam.exe that would be included as a resource to the Bot. So i can extract it at runtime of the bot and specify the exe-name and path where to extract and start it from.
The most important part of that webcam.exe is to add the necessary librarys from OpenCV (www.opencv.org) and try to grab a picture from a connected webcam.
But wait, you could see if your cam would take pictures by an active LED and this could raise attention and suspiciousness on the Bot-PC, right?

I did remember some old tweets on twitter about an article where in the Logitech forums was discussed how to disable the Webcam-LED. It was only a few lines on the Bot and the GUI to add this option as well. It is all about the registry in this case:

Quote from the forums:

> For QuickCam 11.5.0.1169 and above, LVUVC_LEDControl is located in the following registry key:
>
> HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{6BDD1FC6-810F-11D0-BEC7-08002BE2092F}\0000\Settings
>
> Note - If more than one camera is installed, you will have a "folder" for each device (i.e., 0000, 0001, 0002, etc...).
>
> It has a default Data Value of REG_DWORD = 0x00000005 (5).
>
> Based on your comments, I will assume that the following information is true:
>
> 0x00000000 (0) = LED Off
>
> 0x00000008 = LED On

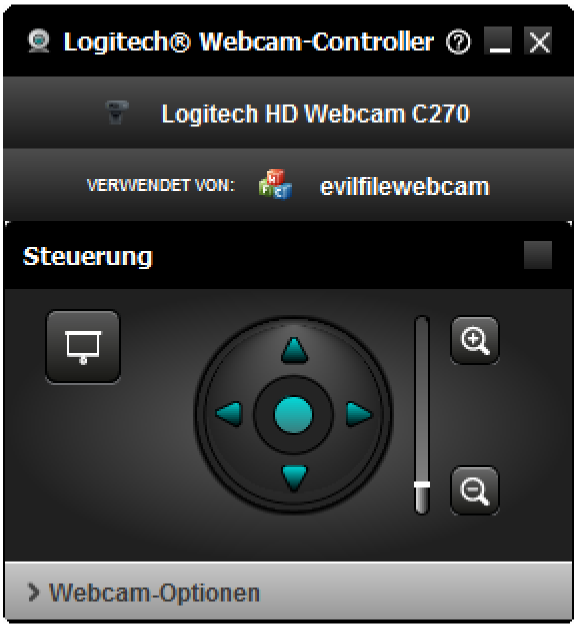
The Bot, if commanded to, would enum all subfolders and add the required RegKey to turn the LED on or off.

Partial code again?

```
if(RegOpenKeyEx(HKEY_LOCAL_MACHINE,(LPCTSTR)"SYSTEM\\CurrentControlSet\\Control\\Class\\{6BDD1FC6-810F-11D0-BEC7-08002BE2092F}\\0000\\Settings" ,
             0,KEY_ALL_ACCESS, &hKey )==ERROR_SUCCESS)
   {
```

```
key.Open( HKEY_LOCAL_MACHINE, "SYS-
TEM\\CurrentControlSet\\Control\\Class\\{6BDD1FC6-810F-11D0-BEC7-
08002BE2092F}\\0000\\Settings", KEY_WRITE);
key.SetValue( x, "LVUVC_LEDControl");
key.Close();

    }
```

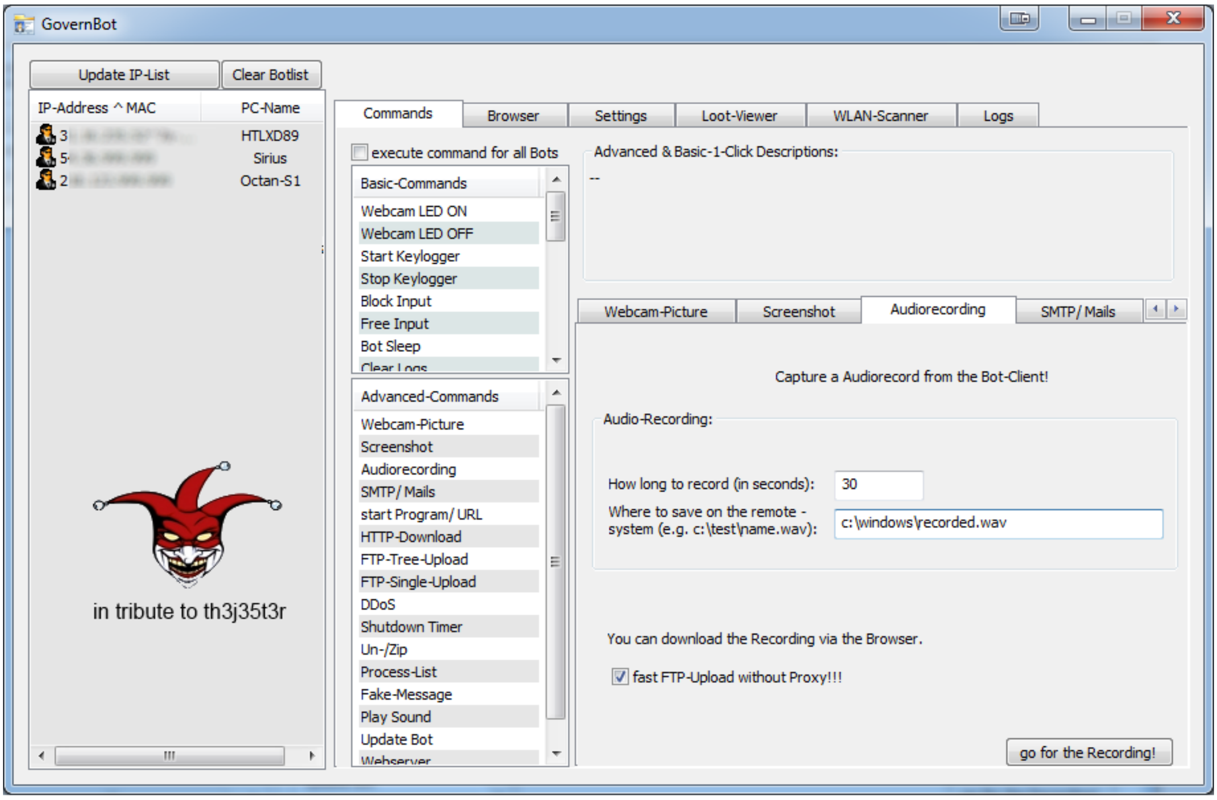
Some of you might know that you could turn on some special features (e.g. with Logitech-Software) to alert you if some application wants to use your webcam. That is why this webcam.exe can be edited by the GUI and given each name you want it to and every possible path to extract it to on the Bot-PC.



VERWENDET VON: evilfilewebcam

After it has done its dirty work, it will selfdelete (even the extracted opencv.dll files) and upload the picture to the attackers ftp-Server (optional).

Pretty evil, no? **That is why you should REALLY unplug your cam if you don’t use it!**



IP-Address ^ MAC	PC-Name
3 [redacted]	HTLXD89
5 [redacted]	Sirius
2 [redacted]	Octan-S1

□ execute command for all Bots

Advanced & Basic-1-Click Descriptions:

--

Capture a Audiorecord from the Bot-Client!

Audio-Recording:

How long to record (in seconds): 30

Where to save on the remote - system (e.g. c:\test\name.wav): c:\windows\recorded.wav

You can download the Recording via the Browser.

☑ fast FTP-Upload without Proxy!!!

If you are using a webcam with a microphone, for Skype and other applications, we could record audio as well – independently from the stolen picture or any webcam feature. We just use the standard recording device and got our way into your living room or where ever your machine is located.

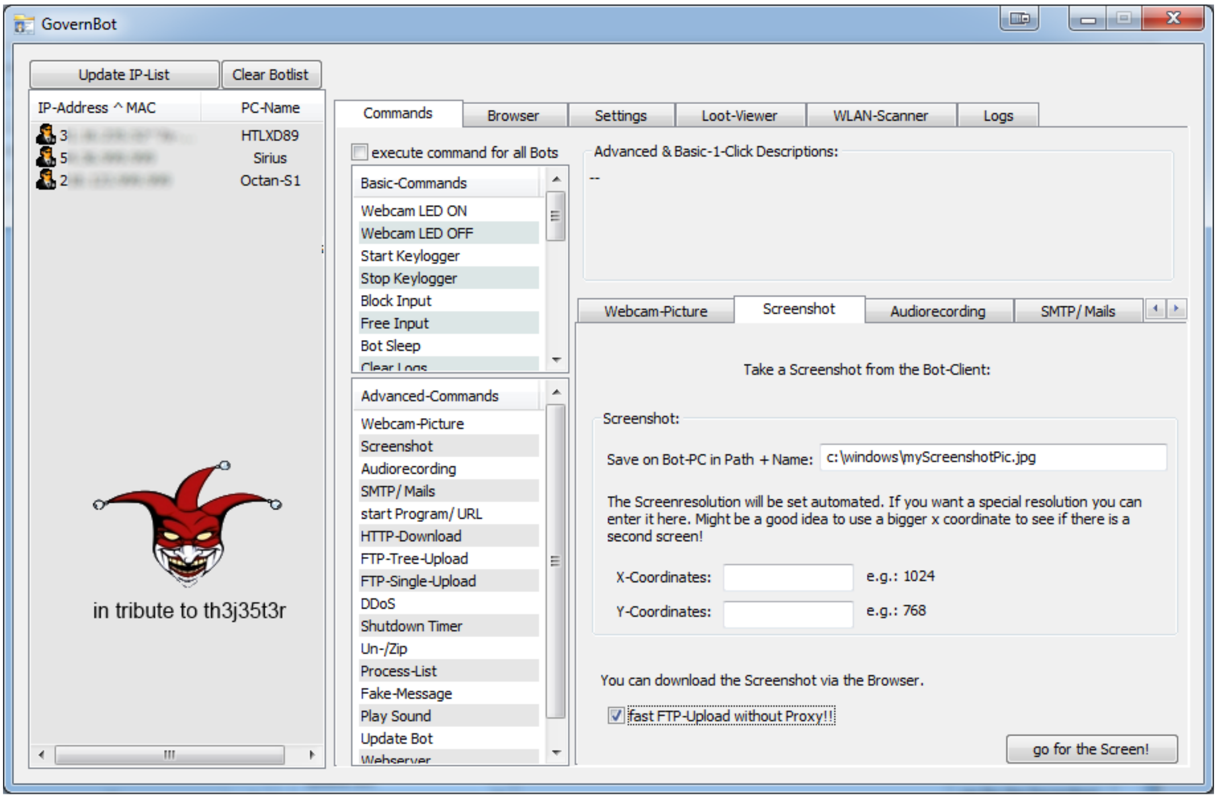
This could work with a TV with webcam/ microphone or your mobilephone as well if coded the right way…

Partial code for our Bot again:

```
    MCI_OPEN_PARMS mciOpenParms;
    MCI_RECORD_PARMS mciRecordParms;
    MCI_SAVE_PARMS mciSaveParms;
    MCI_PLAY_PARMS mciPlayParms;

    mciOpenParms.lpstrDeviceType = "waveaudio";
    mciOpenParms.lpstrElementName = "";
   if (dwReturn = mciSendCommand(0, MCI_OPEN, MCI_OPEN_ELEMENT |
MCI_OPEN_TYPE,(DWORD)(LPVOID) &mciOpenParms))
      {
            return "0";
    }

    wDeviceID = mciOpenParms.wDeviceID;

    mciRecordParms.dwTo = dwMilliSeconds;
    if (dwReturn = mciSendCommand(wDeviceID, MCI_RECORD, MCI_TO | MCI_WAIT
, (DWORD)(LPVOID) &mciRecordParms))
      {
        mciSendCommand(wDeviceID, MCI_CLOSE, (DWORD)NULL, (DWORD)NULL);
                        return "0";

    }
```

Simple known functions you say… yes… but if executed against you it is no fun at all – no matter how simple the coding was! The MSDN is every coders friend. Whatever intentions he has. Good or bad.

Solution: **deactivate or unplug your microphone from the machine if not in use!**



IP-Address ^ MAC	PC-Name
3[redacted]	HTLXD89
5[redacted]	Sirius
2[redacted]	Octan-S1

□ execute command for all Bots

Advanced & Basic-1-Click Descriptions:

--

Take a Screenshot from the Bot-Client:

Screenshot:

Save on Bot-PC in Path + Name: c:\windows\myScreenshotPic.jpg

The Screenresolution will be set automated. If you want a special resolution you can enter it here. Might be a good idea to use a bigger x coordinate to see if there is a second screen!

X-Coordinates: e.g.: 1024

Y-Coordinates: e.g.: 768

You can download the Screenshot via the Browser.

☑ fast FTP-Upload without Proxy!!

Another way to spy at you is to grab some screenshots from your PC and transfer them to the attacker or enable a vnc like stream to the attacker. With a stream he could monitor your actions as they appear on your screen - in real-time.
For this full functional Demo-Bot we will focus on the screenshots taken from your machine, which could capture any given resolution (for 2 or more monitors as well).

There is no further explanation necessary i guess. Let us skip to the partical code that will check for the screen resolution on your machine and that takes the screen silently to create a .jpeg, .png or .bmp:

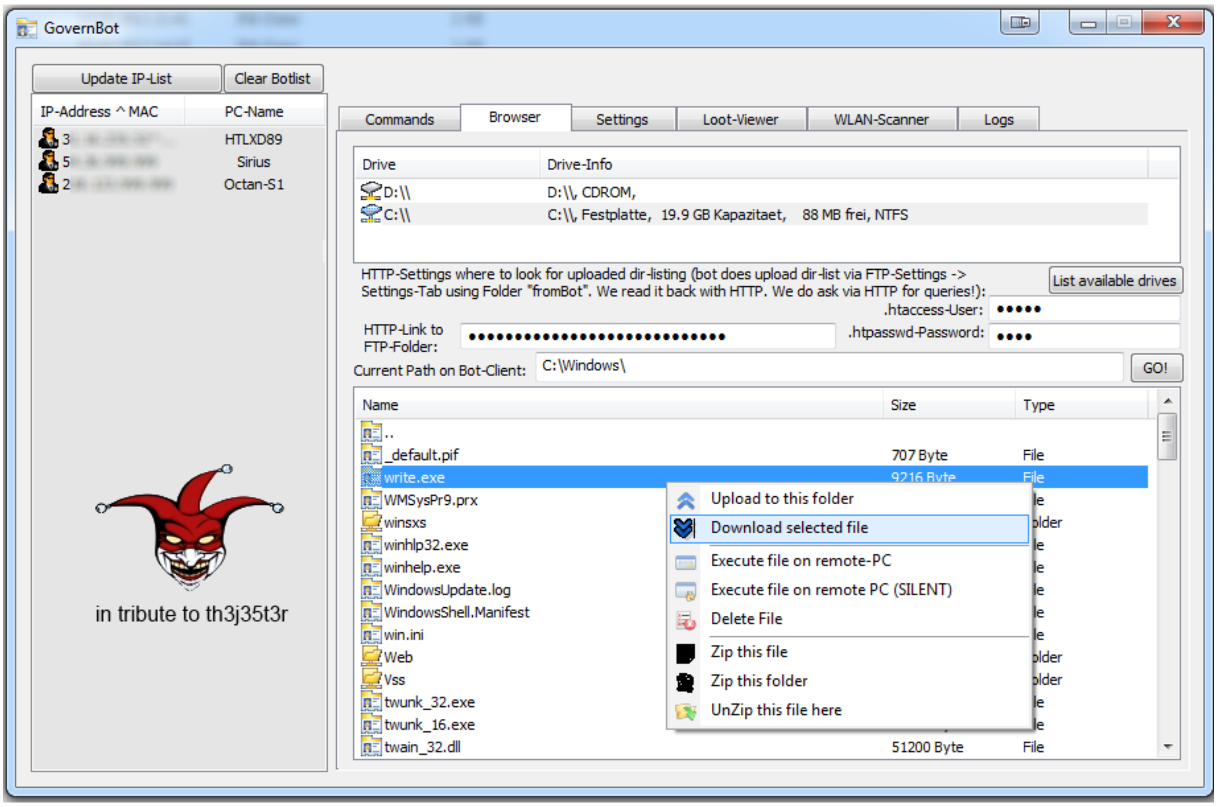
```
   ix = GetSystemMetrics(SM_CXSCREEN);
   iy = GetSystemMetrics(SM_CYSCREEN);

   GetScreenCap(hdc, ix, iy, filename);

 //…

    OldObj=SelectObject(hdc2,aBmp);
    BitBlt(hdc2,0,0,width,height,hdc,0,0,SRCCOPY);

    ZeroMemory(&bmfh,sizeof(BITMAPFILEHEADER));
    bmfh.bfOffBits=sizeof(BITMAPFILEHEADER)+sizeof(BITMAPINFOHEADER);
    bmfh.bfSize=(3*bmih.biHeight*bmih.biWidth)+sizeof(BITMAPFILEHEADER)
                                              +sizeof(BITMAPINFOHEADER);

    bmfh.bfType=0x4d42;
    bmfh.bfReserved1 = 0;
    bmfh.bfReserved2 = 0;
    fileHandle=CreateFile(filename,GENERIC_READ|GENERIC_WRITE,(DWORD)0,NULL,
                                CREATE_ALWAYS,FILE_ATTRIBUTE_NORMAL,(HANDLE) NULL);
//…
```

Stealing your files or manipulating them with ease:

To effectively control a Bot-PC you would want to have the possibility to walk and manipulate its filesystem. This is where i did add a File-Browser that would request a given folder and list all included files and folders. Via right-click context menu you can un-/zip files (optional with password), download, upload new files, Execute them (even silently), or delete them.

This time the GUI will sent a http-request to the http-server that our Bot will read and execute from there. After the Bot knows what to do (e.g. „show me drive c:\") it will enum all files and folders within c:\ recursively and upload the result via ftp to a ftp-Server with the given credentials (in the GUI http-request embedded -> server, user,pass, port, proxy…).
Our Bot will read this result, via a http-request to that uploaded file (basic-auth again to access the file), because we might want to stay anonymous via http-proxy all the time (the GUI has some options for that ;-)

Screenshot – listing path „c:\Windows" on my infected VM:



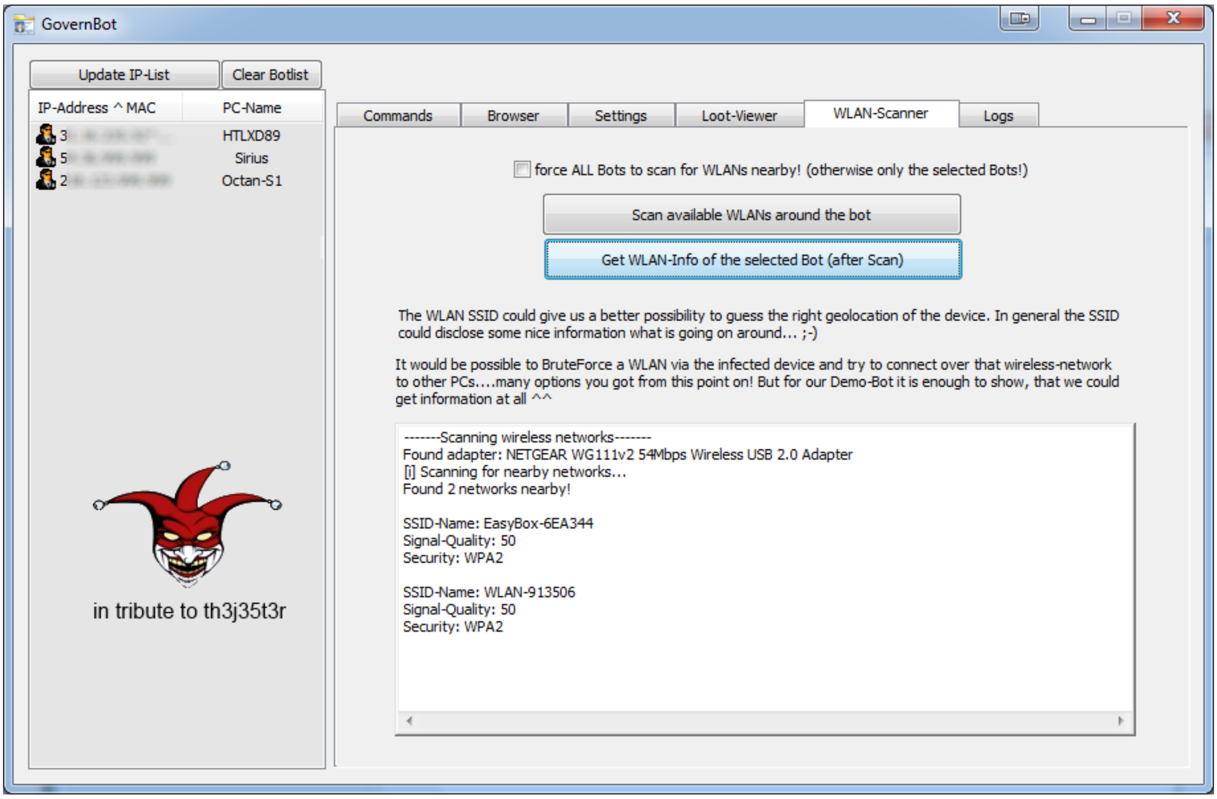
IP-Address ^ MAC	PC-Name
3[redacted]	HTLXD89
5[redacted]	Sirius
2[redacted]	Octan-S1

Drive	Drive-Info
D:\\	D:\\, CDROM,
C:\\	C:\\, Festplatte, 19.9 GB Kapazitaet, 88 MB frei, NTFS

HTTP-Settings where to look for uploaded dir-listing (bot does upload dir-list via FTP-Settings -> Settings-Tab using Folder "fromBot". We read it back with HTTP. We do ask via HTTP for queries!):
.htaccess-User: •••••
HTTP-Link to FTP-Folder: ••••••••••••••••••••••••••••
.htpasswd-Password: ••••
Current Path on Bot-Client: C:\Windows\

Name	Size	Type
..		
_default.pif	707 Byte	File
write.exe	9216 Byte	File
WMSysPr9.prx		le
winsxs		older
winhlp32.exe		le
winhelp.exe		le
WindowsUpdate.log		le
WindowsShell.Manifest		le
win.ini		le
Web		older
Vss		older
twunk_32.exe		le
twunk_16.exe		le
twain_32.dll	51200 Byte	File

Scanning for nearby Wireless Local Area Networks (WLAN):



IP-Address ^ MAC	PC-Name
[redacted]	HTLXD89
[redacted]	Sirius
[redacted]	Octan-S1

□ force ALL Bots to scan for WLANs nearby! (otherwise only the selected Bots!)

The WLAN SSID could give us a better possibility to guess the right geolocation of the device. In general the SSID could disclose some nice information what is going on around... ;-)

It would be possible to BruteForce a WLAN via the infected device and try to connect over that wireless-network to other PCs....many options you got from this point on! But for our Demo-Bot it is enough to show, that we could get information at all ^^

```
-------Scanning wireless networks-------
Found adapter: NETGEAR WG111v2 54Mbps Wireless USB 2.0 Adapter
[i] Scanning for nearby networks...
Found 2 networks nearby!

SSID-Name: EasyBox-6EA344
Signal-Quality: 50
Security: WPA2

SSID-Name: WLAN-913506
Signal-Quality: 50
Security: WPA2
```

I decided to add a scan-option for WLAN to GovernBot, so i am able to scan for WLAN that are close to the infected machines, if it has a WLAN-Adapter installed. It shows us the SSID (Service Set Identifier), Signal-Quality and Security-Mode (such as WEP, WPA or WPA2).

It could be possible to try to BruteForce those found WLAN-AP and let the Bot try to connect to it etc. etc. But i'll stop at the scan here…
With the information about Signal-Quality and SSID you could try to lookup different things (e.g. via Google-Services) to gather more information (Geolocation….) on your target. Be creative ;-)

If you setup your WLAN-AP, be sure to name it somewhat anonymous and use highest encryption.

This bot has much more functions and power which i can not (or in some cases want not) explain in detail to the public. Some are trivial and some are very advanced (like SSH-tunneling from the attacker over http-Server, through the Bot to the SSH-Server). I really hope some guys out there will find this little brief summary usefull. Maybe you got a deeper understanding of how such a piece of software could work and affect you and your privacy.

Be sure to keep an open eye on the running processes and connections on your devices.
The devil is a trickster ;-)

Bye! @RootDial