Hacker-Machine Interface

State of the Union for SCADA HMI Vulnerabilities
Introduction
Trend Micro Zero Day Initiative

• Fritz Sands - @FritzSands
  – Security Researcher – Zero Day Initiative
  – Root cause analysis and vulnerability discovery
  – Focused on SCADA HMI vulnerability analysis

• Brian Gorenc - @maliciousinput
  – Senior Manager - Zero Day Initiative
  – Root cause analysis and vulnerability discovery
  – Organizer of Pwn2Own hacking competitions
SCADA Industry
Marketplace Overview

- Focused on ICS equipment sales over software sales
- Active merger and acquisition activity
- Highly regionalized
What is the Human Machine Interface?

• Main hub for managing and operating control systems
• Collects data from the control systems
• Presents visualization of the system architecture
• Alarms operator/sends notifications
• Should be operated on isolated and trusted networks
  – …but it usually isn’t!
Why target the Human Machine Interface?

• Control the targeted critical infrastructure
• Harvest information about architecture
• Disable or deceive the alarm and notification systems
• Physically damage SCADA equipment
Malware Targeting HMI Solutions

• Stuxnet
  – First malware created to target ICS environments
  – Abused vulnerabilities
    • Siemems SIMATIC STEP 7 DLL Hijacking Vulnerability (ICSA-12-205-02)
    • Siemems WinCC Insecure SQL Server Authentication (ICSA-12-205-01)

• BlackEnergy
  – Ongoing sophisticated malware campaign compromising ICS environments
  – Abused vulnerabilities
    • GE CIMIPCITY Path Traversal Vulnerabilities (ICSA-14-023-01)
    • Siemens WinCC Remote Code Execution Vulnerabilities (ICSA-14-329-02D)
    • Advantech WebAccess (ICS-ALERT-14-281-01B)
ICS-CERT

• Organization within Department of Homeland Security
• Focuses on:
  – Responding to and analyzing control systems-related incidents
  – Conducting vulnerability and malware analysis
  – Providing onsite incident response services
  – Coordinating the responsible disclosure of vulnerabilities and associated mitigations
• For 2015, ICS-CERT responded to 295 incidents and handled 486 vulnerability disclosures
Critical Infrastructure Attacks
Targeting Water Utilities

- Compromised internet-facing AS/400 system responsible for:
  - Network routing
  - Manipulation of Programmable Logic Controllers (PLC)
  - Management of customer PII and billing information
- Altered settings related to water flow and amount of chemicals that went into the water supply
- Four separate connections to the AS/400 over a 60-day period
- Actors IP tied to previous hacktivist activities
Targeting Power Plants

• On December 24, 2015, Ukrainian companies experienced unscheduled power outages impacting 225,000+ customers.
  – Caused by external malicious actors
  – Multiple coordinated attacks within 30 minutes of each other
• Used remote administration tools and/or remote industrial control system (ICS) client software to control breakers.
• Used KillDisk to overwrite Windows-based human-machine interface system.
  – Disrupt restoration efforts
Targeting Railway and Mining Industry

• Malware similar to the power incident found in the attacks against a Ukrainian rail and a Ukrainian mining company
  – November – December 2015

• Overlap between the samples found in the Ukrainian power incident and those apparently used against the Ukrainian mining company
  – Malware leveraged (BlackEnergy/KillDisk)
  – Infrastructure
  – Naming Conventions
Prevalent Vulnerability Types
Current State of HMI Solutions

• Not built with security in mind
• Seen no benefit of the evolution of the secure SDL
• Mitigations against advanced attacks are disabled
  – ASLR, SafeSEH, Stack Cookies
• Poor design/developer assumptions
• Lack of understanding of real operating environment
  – Not on isolated or trusted networks
  – Continually being interconnected
Common Problems with HMI

- Memory Corruption
- Credential Management
- Insecure Defaults
- Authentication/Authorization
- Injection
- Other

Source: 2015-2016 ICS-CERT Advisories
Now let’s get down and dirty!

CASE STUDIES
Injections

- 9% of identified vulnerabilities
- Common vulnerability types
  - SQL Injection
  - Code Injection
  - OS Command Injection
  - Command Injection
- Zero Day Initiative case study
  - Cogent DataHub Gamma Command Injection
  - Remote Code Execution Vulnerability
Cogent DataHub Case Study

• ICS-CERT states:
  – “allow an attacker to turn on an insecure processing mode in the web server, which subsequently allows the attacker to send arbitrary script commands to the server”

• Identifiers
  – CVE-2015-3789
  – ZDI-15-438
  – ICSA–15–246–01

• CVSS
  – 7.5

• Disclosure Timeline
  – 2015–06–02 - Reported to vendor
  – 2015–09–08 – Coordinated release

• Credit
  – Discovered by: Anonymous
  – Disclosed by: Zero Day Initiative
Cogent DataHub Overview
Gamma Script Overview

• Gamma is DataHub’s scripting language
• Dynamically-typed interpreted programming language specifically designed to allow rapid development of control and user interface applications
• Gamma has a syntax similar to C and C++, but has a range of built-in features that make it a far better language for developing sophisticated real-time systems
Attacker-Supplied Script Evaluation

- Flaw exists within the EvalExpression method
  - Allows for execution of attacker controlled code
- Remotely accessible through the AJAX facility
  - Listening on TCP port 80
- Supplying a specially formatted Gamma script allows for the execution of arbitrary OS commands
Vulnerable Code

```java
method AJAXSupport.EvalExpression(expression)
{
    if (allow_any_expression)
    {
        eval (expression);  // Bug here.
    }
    else
    {
        error ("Arbitrary expression evaluation is disabled");
    }
}
```
Exploitation Steps

1. Send a request to any Gamma script to load necessary libraries
2. Call AJAXSupport.AllowExpressions and set allow_any_expression to True
3. Call AJAXSupport.EvalExpression method and pass in the script that you want executed
Patch Analysis

/* Any code to be run when the program gets shut down. */
method AJAXSupport.destructor()
{

method AJAXSupport-AllowExpressions(enable)
{
    .allow_any_expression = (enable == 0 || enable == nil);
}

method AJAXSupport.XMLEscape (str)
{
    local remainder = str;
    local spot = 0;
    while (spot = strchr(remainder, '\')) { --1
        str = string (str, substr(remainder, 0, spot), '\\');
        remainder = substr (remainder, spot + 1, -1);
    }
    str = string (str, remainder);
}

method AJAXSupport-EvalExpression(expression)
{
    if (.allow_any_expression)
    {
        eval (expression);
    }
    else
    {
        error ('Arbitrary expression evaluation is disabled');
    }
}

method AJAXSupport-Test (args...=nil)
{
    string [].XMLHeader, .XMLHeaderSeparator, .XMLVersionString,
    "<test-data name="test" value="01" arg="\"",
    .XMLEscape(string(args), 0); %</test>");
}

/* Any code to be run when the program gets shut down. */
method AJAXSupport.destructor()
{

method AJAXSupport.XMLEscape (str)
{
    local remainder = str;
    local spot = 0;
    while (spot = strchr(remainder, '\')) { --1
        str = string (str, substr(remainder, 0, spot), '\\');
        remainder = substr (remainder, spot + 1, -1);
    }
    str = string (str, remainder);
}

method AJAXSupport-EvalExpression(expression)
{
    if (.allow_any_expression)
    {
        eval (expression);
    }
    else
    {
        error ('Arbitrary expression evaluation is disabled');
    }
}

method AJAXSupport-Test (args...=nil)
{
    string [].XMLHeader, .XMLHeaderSeparator, .XMLVersionString,
    "<test-data name="test" value="01" arg="\"",
    .XMLEscape(string(args), 0); %</test>");
}
Authentication/Authorization

- 12% of identified vulnerabilities
- Common vulnerability types
  - Authentication Bypass Issues
  - Improper Access Control
  - Improper Privilege Management
  - Improper Authentication
- Zero Day Initiative case study
  - Advantech WebAccess upAdminPg Information Disclosure Vulnerability
Advantech WebAccess Case Study

• ICS-CERT states:
  – “A properly authenticated administrator can view passwords for other administrators.”

• Identifiers
  – CVE-2016-5810
  – ZDI-16-429
  – ICSA-16-173-01

• CVSS
  – 6.8

• Disclosure Timeline
  – 2016–05–11 - Reported to vendor
  – 2016–07–18 – Coordinated release

• Credit
  – Discovered by: Zhou Yu
  – Disclosed by: Zero Day Initiative
Update your password...and view others!

- Specific flaw exists within upAdminPg.asp
  - /broadWeb/user/upAdminPg.asp
- Used to update user configurations
  - User Name/Password
  - Description of Account
Easy steps to exploitation

1. Login known account and click ‘Project user property’.
   – http://<ip>/broadWeb/user/upAdminPg.asp?uname=known&return=bwpro

2. Set uname to victim account name.
   – http://<ip>/broadWeb/user/upAdminPg.asp?uname=victim&return=bwproj

3. View the source in page return and examine the password of the user.

```html
<tr bgcolor="#C9C9CB">
   <td width="30%" align="right"><font class=e3>Password</font></td>
   <td width="70%" align="left" valign="middle">
      <input type="password" name="Password" maxlength="8" size="8" value="secret">
   </td>
</tr>
```
Exploitation Demo

Welcome to Advantech WebAccess

Advantech WebAccess is the first full browser-based software package for human-machine interfaces (HMI) and supervisory control and data acquisition (SCADA). It is one of Advantech’s Intelligent Automation Solutions to help integrate all devices in terms of Intelligent Infrastructure and Smart Manufacturing which structure the Advantech Automation framework.

Project Management

admin

Remember me

Back  Login

Copyright © 1983-2014 Advantech Co. Ltd. All Rights Reserved
Insecure Defaults

- 12% of identified vulnerabilities
- Common vulnerability types
  - Cleartext Transmission of Sensitive Information
  - Missing Encryption of Sensitive Info
  - Unsafe ActiveX Control Marked Safe For Scripting
- Zero Day Initiative case study
  - Schneider Electric DS-NVs Rvctl.RVControl.1 SetText Remote Code Execution Vulnerability
Schneider Electric Pelco DS-NVs

- ICS-CERT states:
  - “products contain vulnerable DLL”

- Identifiers
  - CVE-2015-0982
  - ZDI-15-090
  - ICSA-15-071-01

- CVSS
  - 7.5

- Disclosure Timeline
  - 2014–08–13 - Reported to vendor
  - 2015–03–12 – Coordinated release

- Credit
  - Discovered by: rgod and kimiya
  - Disclosed by: Zero Day Initiative
ActiveX Control Settings

Class RVCtrl
GUID: {6781FF2E-7452-11D4-84D4-0040F60CE591}
Number of Interfaces: 7
Default Interface: IRVCtrl
RegKey Safe for Script: False
RegKeySafe for Init: False
KillBitSet: False

View Object Safety Report

Report for Class: {6781FF2E-7452-11D4-84D4-0040F60CE591}
RegKey Safe for Script: False
RegKey Safe for Init: False
Implements IObjectSafety: True
IDispSafe: Safe for untrusted: caller, data
PersistSafe: Safe for untrusted: caller, data
IPSSafe: Safe for untrusted: caller, data
Proof of Concept Demo
Credential Management

- 19% of identified vulnerabilities
- Common vulnerability types
  - Use of Hard-coded Credentials
  - Storing Passwords in a Recoverable Format
  - Insufficiently Protected Credentials
- Zero Day Initiative case study
  - GE MDS PulseNET Hidden Support Account Remote Code Execution Vulnerability
GE MDS PulseNET Case Study

- **ICS-CERT states:**
  - “The affected products contain a hard-coded support account with full privileges.”

- **Identifiers**
  - CVE-2015-6456
  - ZDI-15-440
  - ICSA-15-258-03

- **CVSS**
  - 9.0

- **Disclosure Timeline**
  - 2015–05–14 - Reported to vendor
  - 2015–09–16 – Coordinated release

- **Credit**
  - Discovered by: Andrea Micalizzi (rgod)
  - Disclosed by: Zero Day Initiative
User Management Panel
# Actual User Database

![Database Table](image.png)

<table>
<thead>
<tr>
<th>username</th>
<th>password</th>
<th><em>service</em></th>
<th><em>report</em></th>
<th>phone</th>
<th>email</th>
<th>mobile</th>
<th>page_number</th>
</tr>
</thead>
<tbody>
<tr>
<td>eb593bce-b242-44dc-8bf5-50ce3f37a417</td>
<td>69351b36d8f879d1dbced1bacfd1ca</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7103b69e-48e2-4e7b-903d-f561b20987c</td>
<td>90ac11a1d7f521db734b50d0f50b3c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4f4b7e5e-4f19-11df-5c54-005056000008</td>
<td>admin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9adb2d2df7b4d41d1f6f5000506c00008</td>
<td>operator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c700c557-07cb-4a5b-b67b-1efa88b0c9c</td>
<td>ge_support</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Trend Micro Logo](image.png)
Undocumented ge_support Account

- Exists in the sec_user table *by default*
- Password for this account:
  - ![HDATA[MD5$8af7e0cd2c76d2faa98b71f8ca7923f9
  - “Pu1seNET”
- Account offers full privileges
Exploitation Demo
Memory Corruption

• 20% of identified vulnerabilities

• Common vulnerability types
  – Stack-based Buffer Overflow
  – Heap-based Buffer Overflow
  – Out-of-bounds Read/Write

• Zero Day Initiative case study
  – Advantech WebAccess webvrpcs Service BwOpcSvc.dll WindowName sprintf Stack-Based Buffer Overflow Remote Code Execution Vulnerability
Advantech WebAccess Case Study

• ICS-CERT states:
  – “There are many instances where the buffer on the stack can be overwritten”

• Identifiers
  – CVE-2016-0856
  – ZDI-16-048
  – ICSA-16-014-01

• CVSS
  – 9.3

• Disclosure Timeline
  – 2015–09–17 - Reported to vendor
  – 2016–02–05 – Coordinated release

• Credit
  – Discovered by: Anonymous
  – Disclosed by: Zero Day Initiative
Advantech WebAccess HMI Solution
Remotely Accessible Services

• Launches a service, webvrpcs.exe, in the context of a local administrative users
• Services listens on TCP port 4592, by default, and may be accessed over an RPC-based protocol
• Application interface is structured to resemble the Windows Device IoControl function
  – Each function contains a field similar to an IOCTL
Prototype of RPC function
**IOCTL 0x0001388B**

- Inside BwOpcSvc.dll (which is loaded into webvrpc.exe), routine with an exported entry name of BwSvcFunction which processes a number of entry points, using a jump table.
- Flaw exists within the implementation of IOCTL 0x0001388B
- Stack-based buffer overflow exists in a call to sprintf using WindowsName parameter
Vulnerable Code

```
.text:100015D6  loc_100015D6:  ; CODE XREF:  BwSvcFunction+44j
.text:100015D6                  ; DATA  XREF:  .text:off_10001B080
.text:100015D6      push  offset aRpc_dllBwcmd_g ; jumptable 10001124 case 20011
.text:100015DB      call   sub_10001BB0
.text:100015E0      mov    ebx, [esp+0F8h+arg_8]
.text:100015E7      add    esp, 4
.text:100015EA      lea    edx, [esp+0F4h+WindowName]
.text:100015EE      push   ebx
.text:100015EF      push   offset aS  ; "%s"
.text:100015F4      push   edx       ; char *
.text:100015F5      call   _sprintf
```

Stack Layout

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>text:100010E0</td>
<td>1Param</td>
</tr>
<tr>
<td></td>
<td>= dword ptr -0E4h</td>
</tr>
<tr>
<td>text:100010E0</td>
<td>var_E0</td>
</tr>
<tr>
<td></td>
<td>= dword ptr -0E0h</td>
</tr>
<tr>
<td>text:100010E0</td>
<td>var_DC</td>
</tr>
<tr>
<td></td>
<td>= dword ptr -0DCh</td>
</tr>
<tr>
<td>text:100010E0</td>
<td>var_D8</td>
</tr>
<tr>
<td></td>
<td>= dword ptr -0D8h</td>
</tr>
<tr>
<td>text:100010E0</td>
<td>wParam</td>
</tr>
<tr>
<td></td>
<td>= dword ptr -0D4h</td>
</tr>
<tr>
<td>text:100010E0</td>
<td>var_D0</td>
</tr>
<tr>
<td></td>
<td>= dword ptr -0D0h</td>
</tr>
<tr>
<td>text:100010E0</td>
<td>var_CC</td>
</tr>
<tr>
<td></td>
<td>= byte ptr -0CCh</td>
</tr>
<tr>
<td>text:100010E0</td>
<td>var_C0</td>
</tr>
<tr>
<td></td>
<td>= byte ptr -0C0h</td>
</tr>
<tr>
<td>text:100010E0</td>
<td>File</td>
</tr>
<tr>
<td></td>
<td>= byte ptr -0A0h</td>
</tr>
<tr>
<td>text:100010E0</td>
<td>WindowName</td>
</tr>
<tr>
<td></td>
<td>= byte ptr -080h</td>
</tr>
<tr>
<td>text:100010E0</td>
<td>arg_0</td>
</tr>
<tr>
<td></td>
<td>= dword ptr 4</td>
</tr>
<tr>
<td>text:100010E0</td>
<td>arg_8</td>
</tr>
<tr>
<td></td>
<td>= dword ptr 0Ch</td>
</tr>
<tr>
<td>text:100010E0</td>
<td>arg_C</td>
</tr>
<tr>
<td></td>
<td>= dword ptr 10h</td>
</tr>
<tr>
<td>text:100010E0</td>
<td>arg_10</td>
</tr>
<tr>
<td></td>
<td>= dword ptr 14h</td>
</tr>
</tbody>
</table>
Application Crash

ModLoad: 6fa00000 6fa51000 C:\Windows\SysWOW64\WINSPool.DRV
(2dfe.3be4): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\W
indows\SysWOW64\ntdll.dll -
eax=ffffffff ebx=00000000 ecx=775e38ca edx=00231078 esi=000004e2b edi=06ff0000
eip=7f7f7f7f esp=0909f6f0 ebp=0909f9a4 iopl=0 n v up ei pl nz na po nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b
efl=00010202
7f7f7f7f ?? ?? ??
0:009> kv
ChildEBP RetAddr Args to Child
WARNING: Frame IP not in any known module. Following frames may be wrong.
0909f6ec 7f7f7f7f 00000000 097acc18 00000004 0x7f7f7f7f
*** WARNING: Unable to verify checksum for C:\WebAccess\Node\webvrpcs.exe
*** ERROR: Module load completed but symbols could not be loaded for C:\WebAccess
s\Node\webvrpcs.exe
0909f9a4 0402bb5 0998ee08 0a78cfc0 0001380b 0x7f7f7f7f
0909f9f0 0401198 0998ee08 0a78cfc0 0001380b webvrpcs+0x2bb5
Patch Analysis

• _sprintf is in the list of Microsoft banned APIs list
  – First published in 2007

• Advantech should implement Microsoft banned APIs and remove all of them from shipping code

• What did they do...
Patch Analysis

- WindowName field in the stack buffer is 0x80 bytes
- _snprintf Length parameter is 0x7f bytes

```assembly
.text:10001600  lea   edx, [esp+0F4h+WindowName]
.text:10001604  push  ebx
.text:10001605  push  offset aS ; "%s"
.text:1000160A  push  7Fh   ; size_t
.text:1000160C  push  edx    ; char *
.text:1000160D  call  __snprintf
```
## Variant Analysis

1. ZDI-16-049 - Advantech WebAccess webvrpcs Service BwOpSvc.dll WindowName sprintf Stack-Based Buffer Overflow Remote Code Execution Vulnerability
2. ZDI-16-050 - Advantech WebAccess webvrpcs Service BwOpSvc.dll WindowName sprintf Stack-Based Buffer Overflow Remote Code Execution Vulnerability
3. ZDI-16-051 - Advantech WebAccess webvrpcs Service BwOpSvc.dll WindowName sprintf Stack-Based Buffer Overflow Remote Code Execution Vulnerability
5. ZDI-16-053 - Advantech WebAccess webvrpcs Service BwASScdDl.dll TargetHost strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
6. ZDI-16-054 - Advantech WebAccess webvrpcs Service WaDBS.dll TagName strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
7. ZDI-16-055 - Advantech WebAccess webvrpcs Service Bwpalarm.dll sprintf Stack-Based Buffer Overflow Remote Code Execution Vulnerability
8. ZDI-16-056 - Advantech WebAccess webvrpcs Service Bwpalarm.dll sprintf Stack-Based Buffer Overflow Remote Code Execution Vulnerability
9. ZDI-16-057 - Advantech WebAccess webvrpcs Service Bwpalarm.dll ProjectName strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
10. ZDI-16-058 - Advantech WebAccess webvrpcs Service Bwpalarm.dll ProjectName strcpy Globals Overflow Remote Code Execution Vulnerability
11. ZDI-16-059 - Advantech WebAccess webvrpcs Service Bwpalarm.dll ProjectName strcat Stack-Based Buffer Overflow Remote Code Execution Vulnerability
12. ZDI-16-060 - Advantech WebAccess webvrpcs Service Bwpalarm.dll HostName/ProjectName/NodeName strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
13. ZDI-16-061 - Advantech WebAccess webvrpcs Service Bwpalarm.dll sprintf Stack-Based Buffer Overflow Remote Code Execution Vulnerability
14. ZDI-16-062 - Advantech WebAccess webvrpcs Service Bwpalarm.dll ProjectName/NodeName sprintf Stack-Based Buffer Overflow Remote Code Execution Vulnerability
15. ZDI-16-063 - Advantech WebAccess webvrpcs Service Bwpalarm.dll strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
17. ZDI-16-065 - Advantech WebAccess webvrpcs Service Bwpalarm.dll strcpy Heap-Based Buffer Overflow Remote Code Execution Vulnerability
18. ZDI-16-066 - Advantech WebAccess webvrpcs Service Bwpalarm.dll strcpy Heap-Based Buffer Overflow Remote Code Execution Vulnerability
19. ZDI-16-067 - Advantech WebAccess webvrpcs Service Bwpalarm.dll Backup RPC Hostname strcpy Heap-Based Buffer Overflow Remote Code Execution Vulnerability
20. ZDI-16-068 - Advantech WebAccess webvrpcs Service Bwpalarm.dll strcpy Heap-Based Buffer Overflow Remote Code Execution Vulnerability
22. ZDI-16-070 - Advantech WebAccess webvrpcs Service Bwpalarm.dll Primary RPC Hostname strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
23. ZDI-16-071 - Advantech WebAccess webvrpcs Service Bwpalarm.dll strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
24. ZDI-16-072 - Advantech WebAccess webvrpcs Service Bwpalarm.dll Backup RPC Hostname strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
Variant Analysis

25. ZDI-16-073 - Advantech WebAccess webvrpcs Service BwpAlarm.dll memcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
27. ZDI-16-075 - Advantech WebAccess webvrpcs Service BwpAlarm.dll memcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
29. ZDI-16-077 - Advantech WebAccess webvrpcs Service ViewSrv.dll strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
30. ZDI-16-078 - Advantech WebAccess webvrpcs Service ViewSrv.dll strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
31. ZDI-16-079 - Advantech WebAccess webvrpcs Service ViewSrv.dll strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
32. ZDI-16-080 - Advantech WebAccess webvrpcs Service ViewSrv.dll TagName strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
33. ZDI-16-081 - Advantech WebAccess webvrpcs Service BwKrlApi.dll strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
34. ZDI-16-082 - Advantech WebAccess webvrpcs Service ViewSrv.dll Path BwBuildPath strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
35. ZDI-16-083 - Advantech WebAccess webvrpcs Service ViewSrv.dll Path BwBuildPath Stack-Based Buffer Overflow Remote Code Execution Vulnerability
36. ZDI-16-084 - Advantech WebAccess webvrpcs Service ViewSrv.dll Path BwBuildPath Stack-Based Buffer Overflow Remote Code Execution Vulnerability
37. ZDI-16-085 - Advantech WebAccess webvrpcs Service ViewSrv.dll Path BwBuildPath Stack-Based Buffer Overflow Remote Code Execution Vulnerability
38. ZDI-16-086 - Advantech WebAccess webvrpcs Service BwKrlApi.dll strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
39. ZDI-16-087 - Advantech WebAccess webvrpcs Service BwKrlApi.dll strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
40. ZDI-16-088 - Advantech WebAccess webvrpcs Service BwKrlApi.dll strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
41. ZDI-16-089 - Advantech WebAccess webvrpcs Service BwKrlApi.dll strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
42. ZDI-16-090 - Advantech WebAccess webvrpcs Service BwKrlApi.dll strcpy TagGroup strcat Stack-Based Buffer Overflow Remote Code Execution Vulnerability
43. ZDI-16-091 - Advantech WebAccess webvrpcs Service BwKrlApi.dll strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
44. ZDI-16-092 - Advantech WebAccess webvrpcs Service BwKrlApi.dll Path BwBuildPath Stack-Based Buffer Overflow Remote Code Execution Vulnerability
45. ZDI-16-093 - Advantech WebAccess webvrpcs Service DrawSrv.dll Path BwBuildPath Stack-Based Buffer Overflow Remote Code Execution Vulnerability
46. ZDI-16-094 - Advantech WebAccess webvrpcs Service DrawSrv.dll Path BwBuildPath Stack-Based Buffer Overflow Remote Code Execution Vulnerability
47. ZDI-16-095 - Advantech WebAccess webvrpcs Service DrawSrv.dll TagGroup strcat Stack-Based Buffer Overflow Remote Code Execution Vulnerability
48. ZDI-16-096 - Advantech WebAccess webvrpcs Service ViewDll.dll TagGroup strcat Stack-Based Buffer Overflow Remote Code Execution Vulnerability
Variant Analysis

49. ZDI-16-097 - Advantech WebAccess webvpcs Service View01.dll TagGroup strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
50. ZDI-16-099 - Advantech WebAccess webvpcs Service DrawSrv.dll TagGroup strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
51. ZDI-16-100 - Advantech WebAccess webvpcs Service DrawSrv.dll TagGroup strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
52. ZDI-16-101 - Advantech WebAccess datacore Service datacore.exe Path strcat Stack-Based Buffer Overflow Remote Code Execution Vulnerability
53. ZDI-16-102 - Advantech WebAccess datacore Service datacore.exe Path strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
54. ZDI-16-103 - Advantech WebAccess datacore Service datacore.exe Path strcat Stack-Based Buffer Overflow Remote Code Execution Vulnerability
55. ZDI-16-104 - Advantech WebAccess datacore Service datacore.exe ExtDataSize Integer Overflow Remote Code Execution Vulnerability
56. ZDI-16-105 - Advantech WebAccess datacore Service datacore.exe strcpy Shared Virtual Memory Overflow Remote Code Execution Vulnerability
57. ZDI-16-106 - Advantech WebAccess datacore Service datacore.exe sprintf Stack-Based Buffer Overflow Remote Code Execution Vulnerability
58. ZDI-16-107 - Advantech WebAccess datacore Service datacore.exe strcpy Heap-Based Buffer Overflow Remote Code Execution Vulnerability
59. ZDI-16-108 - Advantech WebAccess datacore Service datacore.exe Username strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
60. ZDI-16-109 - Advantech WebAccess datacore Service datacore.exe strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
61. ZDI-16-110 - Advantech WebAccess datacore Service datacore.exe strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
62. ZDI-16-111 - Advantech WebAccess datacore Service datacore.exe strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
63. ZDI-16-112 - Advantech WebAccess datacore Service datacore.exe Username strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
64. ZDI-16-113 - Advantech WebAccess datacore Service datacore.exe Username strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
65. ZDI-16-114 - Advantech WebAccess datacore Service datacore.exe Username strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
66. ZDI-16-115 - Advantech WebAccess datacore Service datacore.exe strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
67. ZDI-16-116 - Advantech WebAccess datacore Service datacore.exe strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
68. ZDI-16-117 - Advantech WebAccess datacore Service datacore.exe Username strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
69. ZDI-16-118 - Advantech WebAccess datacore Service datacore.exe strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
70. ZDI-16-119 - Advantech WebAccess datacore Service datacore.exe AlarmMessage strcpy Heap-Based Buffer Overflow Remote Code Execution Vulnerability
71. ZDI-16-120 - Advantech WebAccess datacore Service datacore.exe AlarmMessage sprintf Stack-Based Buffer Overflow Remote Code Execution Vulnerability
72. ZDI-16-121 - Advantech WebAccess datacore Service datacore.exe AlarmMessage strcpy Heap-Based Buffer Overflow Remote Code Execution Vulnerability
Researcher Guidance
Basic Fuzzing

- Simple bit-flipping fuzzing is highly effective against HMI
  - Look for new file associations during installations
- Don’t forget to enable page heap to find heap corruption
  - gflags.exe /i hmi.exe +hpa +ust
- Leverage existing tools and frameworks
  - radamsa
  - sqlmap
Microsoft’s Attack Surface Analyzer

- Released in 2012
- Creates snapshots before and after installation
- Highlights security misconfigurations
  - Registry settings and file permissions
- Provides a list of auditable system modifications
  - COM objects
  - ActiveX controls
  - File associations
  - RPC endpoints
Attack Surface Analyzer Report

**Attack Surface Report: Table Of Contents**

- System Information
  - Running Processes
  - Executable Memory Pages
  - Windows
  - Kernel Objects
  - Modules
- Service Information
  - Services
  - Drivers
- ActiveX, DCOM, COM, File Extensions
  - COM Controls
  - DCOM Controls
  - File Registrations
- Internet Explorer
  - IE Preapproved Controls
- Network Information
  - Network Ports
  - Named Pipes
  - RPC Endpoints
- Firewall
  - Firewall Rules
- System Environment, Users, Groups
  - Groups

**Network Information**

<table>
<thead>
<tr>
<th>Type</th>
<th>TCP</th>
<th>UDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>All New Ports (37 total)</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>Running as System</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Running as Local Service</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Running as Network Service</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Running as Other</td>
<td>12</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Port Name</th>
<th>State</th>
<th>Process</th>
<th>Account</th>
</tr>
</thead>
<tbody>
<tr>
<td>5355/UDP -- Unknown Protocol</td>
<td>Unknown</td>
<td>svchost.exe (992)</td>
<td></td>
</tr>
<tr>
<td>5355/UDP -- Unknown Protocol</td>
<td>Unknown</td>
<td>svchost.exe (992)</td>
<td></td>
</tr>
<tr>
<td>4592/TCP -- Unknown Protocol</td>
<td>Listen</td>
<td>webvpcs.exe (1992)</td>
<td>WIN-PC1TQD11C8ZDI</td>
</tr>
<tr>
<td>4592/TCP -- Unknown Protocol</td>
<td>Established</td>
<td>webvpcs.exe (1992)</td>
<td>WIN-PC1TQD11C8ZDI</td>
</tr>
<tr>
<td>1355/TCP -- Unknown Protocol</td>
<td>Established</td>
<td>webvpcs.exe (1992)</td>
<td>WIN-PC1TQD11C8ZDI</td>
</tr>
</tbody>
</table>
# Attack Surface Analyzer Report

## Directories Containing Objects With Weak ACLs

The folder C:\inetpub\wwwroot contains files and/or folders with ACLs that allow tampering by multiple non-administrator accounts.

### Description:

The folder C:\inetpub\wwwroot contains files and/or folders with ACLs that allow tampering by multiple non-administrator accounts.

### Details:

Folder: C:\inetpub\wwwroot  
Contents with bad ACLs:  
1. C:\inetpub\wwwroot\BWDefault.asp  
2. C:\inetpub\wwwroot\Default.asp  
3. C:\inetpub\wwwroot\iisstart.htm

<table>
<thead>
<tr>
<th>Account</th>
<th>Rights</th>
</tr>
</thead>
<tbody>
<tr>
<td>World (S-1-1-0)</td>
<td>WRITE_OWNER WRITE_DAC FILE_WRITE_ATTRIBUTES FILE_WRITE_DATA FILE_APPEND_DATA FILE_WRITE_DATA</td>
</tr>
</tbody>
</table>

### Action:

The ACL should be tightened. Do not allow users to write to start points, files or directories that influence control over other users.
Audit for Banned APIs

- C runtime has many APIs with serious security programs
- Microsoft banned use of problematic C library functions
  - “The Security Development Lifecycle” (Microsoft, 2006)
  - Security Development Lifecycle Banned Function Calls
- Depressingly common in HMI code, with predictable negative impacts
- IDA is extremely valuable tool for auditing for inappropriate uses
Disclosure Statistics
Vulnerability Exposure Windows
Vendor Response Times
Industry by Industry Comparison
Conclusions
Go find bugs!

- ICS-focused malware actively exploiting HMI vulnerabilities
- HMI codebases plagued with critical vulnerabilities
- Simple techniques can be used to find vulnerabilities
- Exposure windows is ~150 days leaving critical infrastructure vulnerable

https://github.com/thezdi
  - Vulnerability White Papers
  - Proofs of Concept
  - Disclosure Data
Questions?

www.zerodayinitiative.com

@thezdi