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*

TREND MICRO

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

Why target the Human Machine Interface?

- Control the targeted critical infrastructure
- Harvest information about architecture
- Disable alarming and notification systems
- Physically damage SCADA equipment

Malware Targeting HMI Solutions

- Stuxnet
  - First malware created to target ICS environments
  - Abused HMI vulnerabilities
    - Siemems SIMATIC STEP 7 DLL Hijacking Vulnerability (ICSA-12-205-02)
    - Siemens WinCC Insecure SQL Server Authentication (ICSA-12-205-01)
- BlackEnergy
  - Ongoing sophisticated malware campaign compromising ICS environments
  - Abused HMI 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

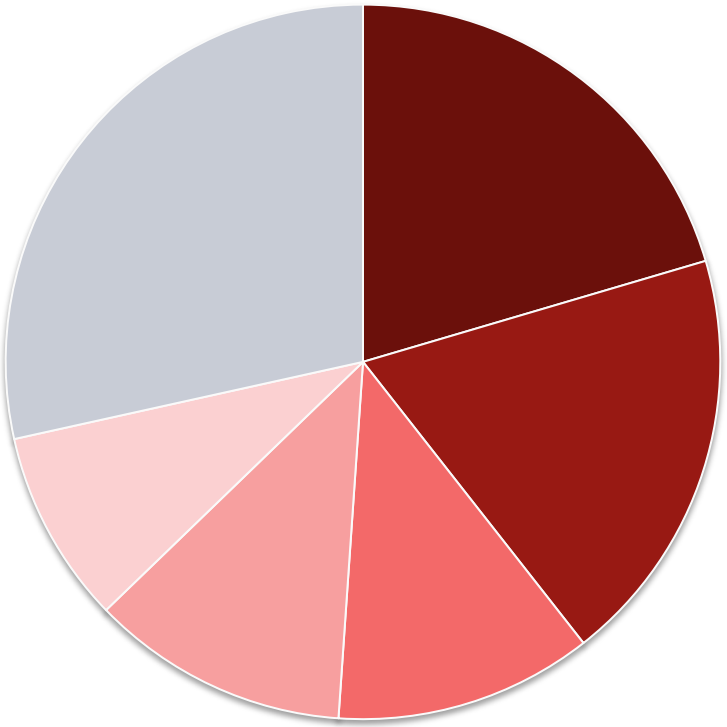
- Malware similar to the power incident found in the attacks against a Ukrainian rail and a Ukrainian mining company
  - November – December 2015
- Overlap between the samples found in the Ukrainian power incident and those apparently used against the Ukrainian mining company
  - Malware leveraged (BlackEnergy/KillDisk)
  - 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
- Poor design/developer assumptions
- Lack of understanding of real operating environment
  - Not on isolated or trusted networks
  - Continually being interconnected

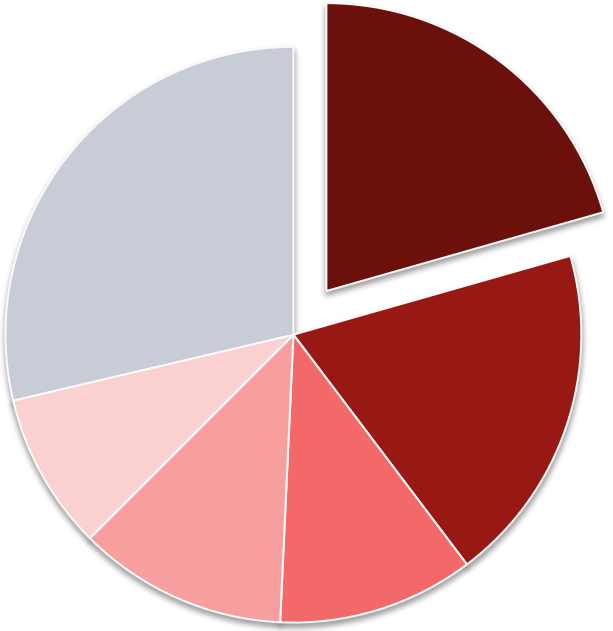
Common Problems with HMI



Source: 2015-2016 ICS-CERT Advisories

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

```
00000154  05 00 00 03 10 00 00 00  c0 00 00 00 05 00 00 00 ........ ........
00000164  a8 00 00 00 00 00 00 00  c0 3f 58 05 8b 38 01 00 ........ .?X..8..
00000174  8c 00 00 00 8c 00 00 00  7f 7f 7f 7f 7f 7f 7f 7f ........ ........
00000184  7f 7f 7f 7f 7f 7f 7f 7f  7f 7f 7f 7f 7f 7f 7f 7f ........ ........
00000194  7f 7f 7f 7f 7f 7f 7f 7f  7f 7f 7f 7f 7f 7f 7f 7f ........ ........
000001A4  7f 7f 7f 7f 7f 7f 7f 7f  7f 7f 7f 7f 7f 7f 7f 7f ........ ........
000001B4  7f 7f 7f 7f 7f 7f 7f 7f  7f 7f 7f 7f 7f 7f 7f 7f ........ ........
000001C4  7f 7f 7f 7f 7f 7f 7f 7f  7f 7f 7f 7f 7f 7f 7f 7f ........ ........
000001D4  7f 7f 7f 7f 7f 7f 7f 7f  7f 7f 7f 7f 7f 7f 7f 7f ........ ........
000001E4  7f 7f 7f 7f 7f 7f 7f 7f  7f 7f 7f 7f 7f 7f 7f 7f ........ ........
000001F4  7f 7f 7f 7f 7f 7f 7f 7f  7f 7f 7f 7f 7f 7f 7f 7f ........ ........
00000204  00 00 00 00 04 00 00 00  04 00 00 00 00 00 00 00 ........ ........
```

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_10001B08o
.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
```

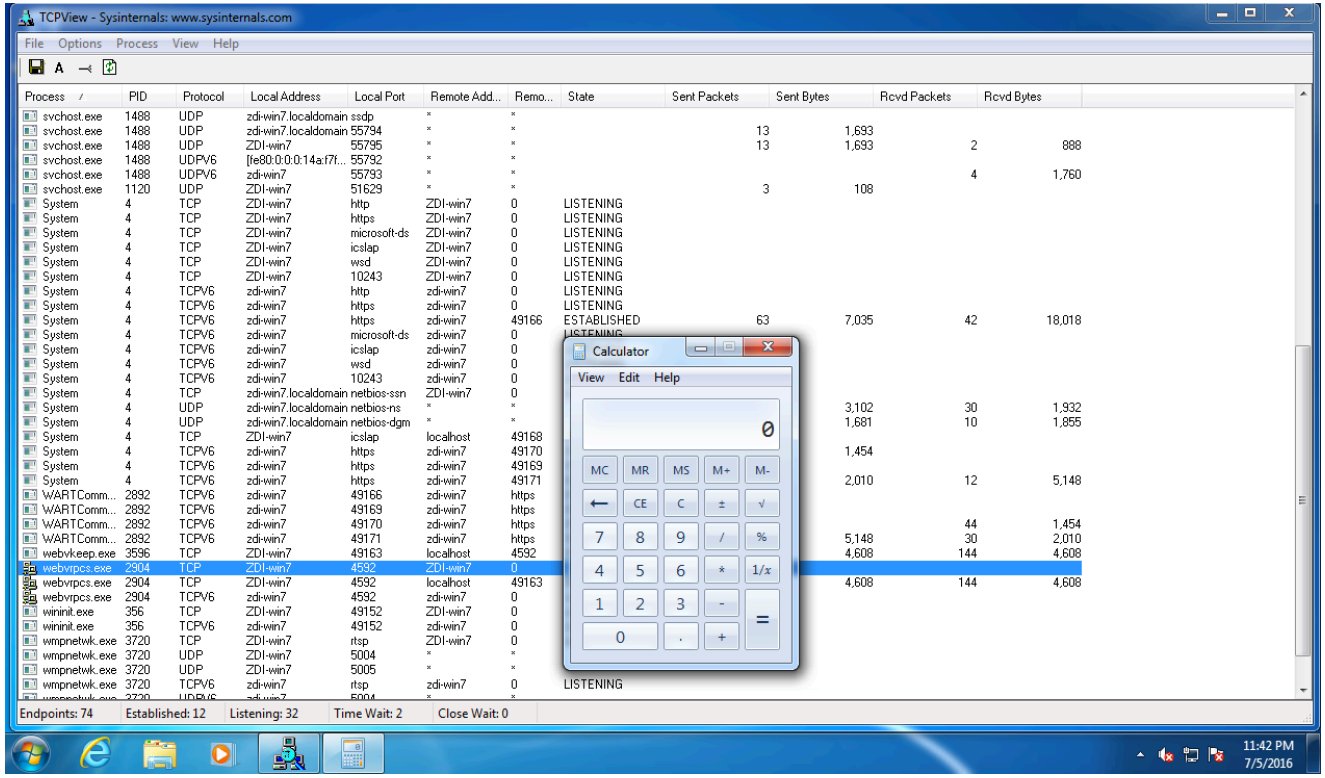
Stack Layout

```
.text:100010E0 lParam          = dword ptr -0E4h
.text:100010E0 var_E0          = dword ptr -0E0h
.text:100010E0 var_DC          = dword ptr -0DCh
.text:100010E0 var_D8          = dword ptr -0D8h
.text:100010E0 wParam          = dword ptr -0D4h
.text:100010E0 var_D0          = dword ptr -0D0h
.text:100010E0 var_CC          = byte ptr -0CCh
.text:100010E0 var_C0          = byte ptr -0C0h
.text:100010E0 File            = byte ptr -0A0h
.text:100010E0 WindowName      = byte ptr -80h
.text:100010E0 arg_0           = dword ptr  4
.text:100010E0 arg_8           = dword ptr  0Ch
.text:100010E0 arg_C           = dword ptr  10h
.text:100010E0 arg_10          = dword ptr  14h
```

Application Crash

```
ModLoad: 6fa00000 6fa51000   C:\Windows\SysWOW64\WINSPOOL.DRV
(2dfc.3bd4): 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=00004e2b edi=06ff0000
eip=7f7f7f7f esp=0909f6f0 ebp=0909f9a4 iopl=0         nv 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:\WebAcces
s\Node\webvrpcs.exe
0909f9a4 00402bb5 0998ee08 0a78cfc0 0001388b 0x7f7f7f7f
0909f9f0 00401198 0998ee08 0a78cfc0 0001388b webvrpcs+0x2bb5
```

Exploitation Demo



Patch Analysis

- _sprintf is in the list of Microsoft banned APIs list
  - First published in 2007
  - https://msdn.microsoft.com/en-us/library/bb288454.aspx
- 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

```
.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 BwOpcSvc.dll WindowName sprintf Stack-Based Buffer Overflow Remote Code Execution Vulnerability
2. ZDI-16-050 - Advantech WebAccess webvrpcs Service BwOpcSvc.dll WindowName sprintf Stack-Based Buffer Overflow Remote Code Execution Vulnerability
3. ZDI-16-051 - Advantech WebAccess webvrpcs Service BwOpcSvc.dll WindowName sprintf Stack-Based Buffer Overflow Remote Code Execution Vulnerability
4. ZDI-16-052 - Advantech WebAccess webvrpcs Service BwOpcSvc.dll sprintf Uncontrolled Format String Remote Code Execution Vulnerability
5. ZDI-16-053 - Advantech WebAccess webvrpcs Service BwBASScdDl.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
16. ZDI-16-064 - Advantech WebAccess webvrpcs Service BwpAlarm.dll strcpy Heap-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
21. ZDI-16-069 - Advantech WebAccess webvrpcs Service BwpAlarm.dll NewPointValue strcpy Stack-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

TREND MICRO

Variant Analysis

25. ZDI-16-073 - Advantech WebAccess webvrpcs Service BwpAlarm.dll memcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
26. ZDI-16-074 - Advantech WebAccess webvrpcs Service BwpAlarm.dll memcpy Globals Overflow Remote Code Execution Vulnerability
27. ZDI-16-075 - Advantech WebAccess webvrpcs Service BwpAlarm.dll memcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
28. ZDI-16-076 - Advantech WebAccess webvrpcs Service ViewSrv.dll strcpy 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 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 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 strcpy 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

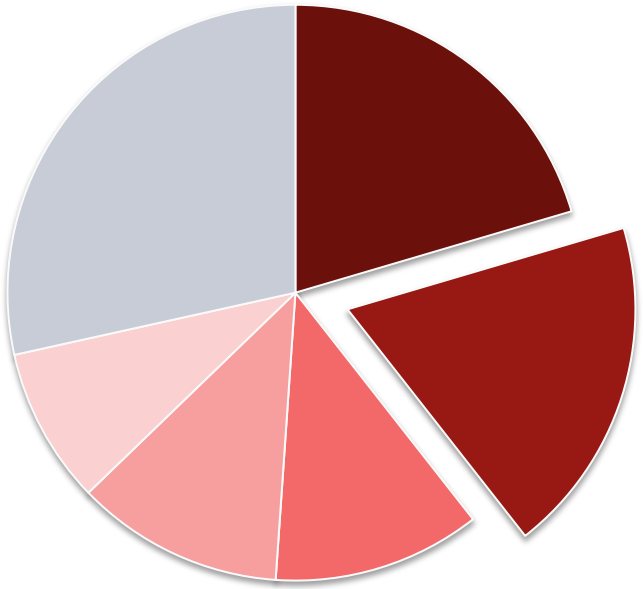
Variant Analysis

49. ZDI-16-097 - Advantech WebAccess webvrpcs Service ViewDll.dll TagGroup strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
50. ZDI-16-099 - Advantech WebAccess webvrpcs Service DrawSrv.dll TagGroup strcpy Stack-Based Buffer Overflow Remote Code Execution Vulnerability
51. ZDI-16-100 - Advantech WebAccess webvrpcs 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 strncpy 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

TREND MICRO

Credential Management

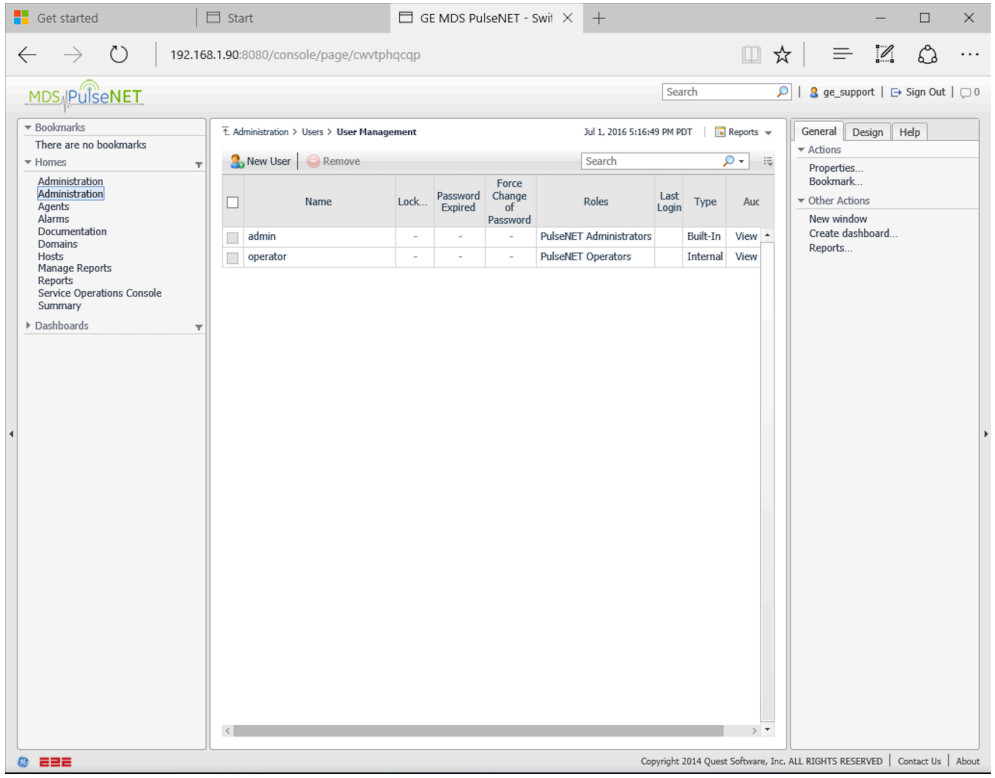
- 19% of identified vulnerabilities
- Common vulnerability types
  - Use of Hard-coded Credentials
  - Storing Passwords in a Recoverable Format
  - Insufficiently Protected Credentials
- Zero Day Initiative case study
  - GE MDS PulseNET Hidden Support Account Remote Code Execution Vulnerability



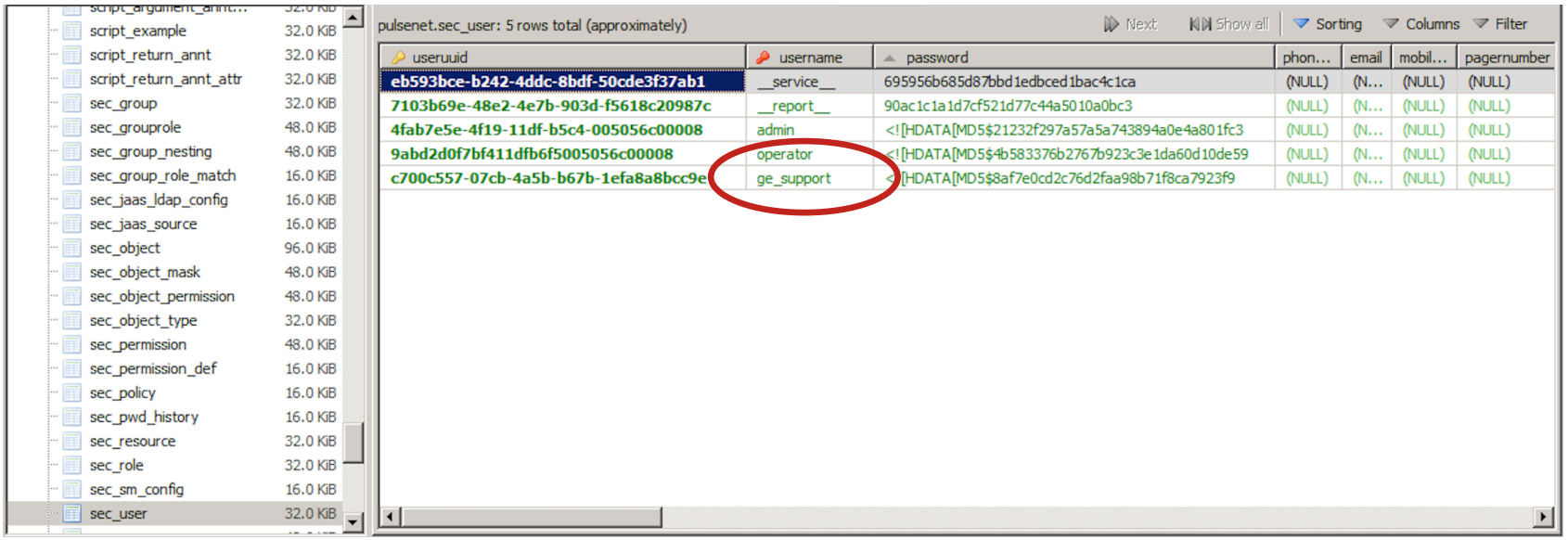
GE MDS PulseNET Case Study

- 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

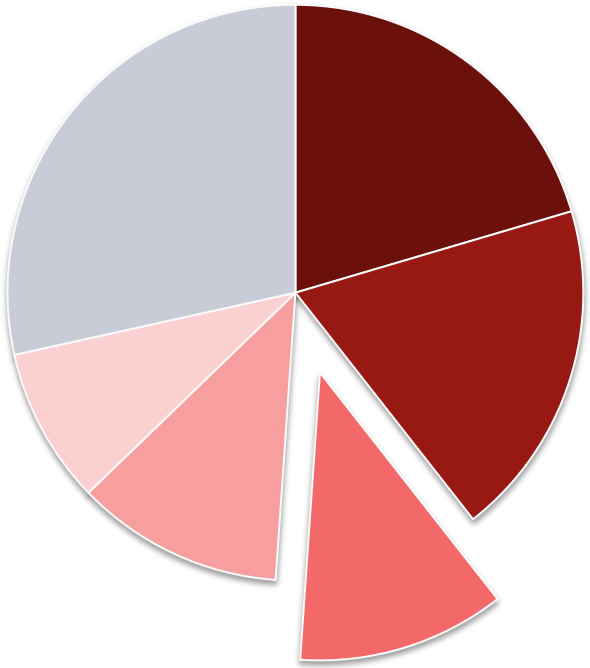


Undocumented ge_support Account

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

Insecure Default

- 12% of identified vulnerabilities
- Common vulnerability types
  - Cleartext Transmission of Sensitive Information
  - Missing Encryption of Sensitive
  - Unsafe ActiveX Control Marked Safe For Scripting
- Zero Day Initiative case study
  - Seimens Case Study

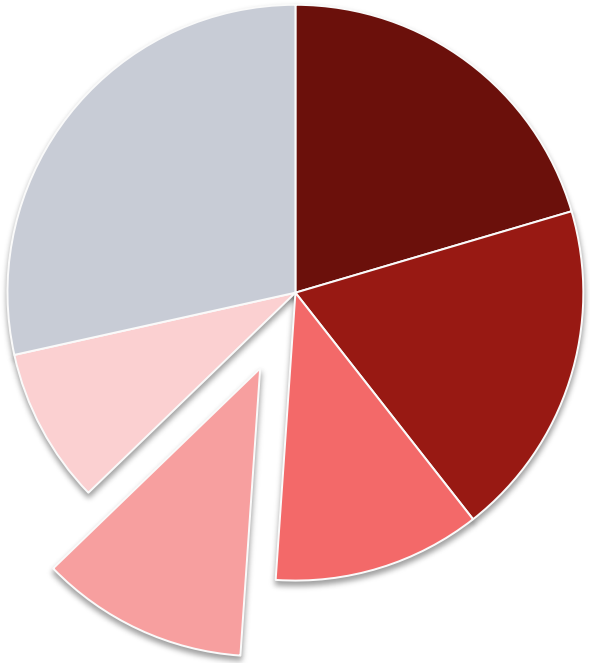


0-day Vulnerability Case Study

- Vulnerability details will be disclosed during the talk at the DEF CON conference
- Expected to patch the week before the conference
- If it is not patched, we will release the details publically in accordance with the Zero Day Initiative Vulnerability Disclosure Policy

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 Case Study

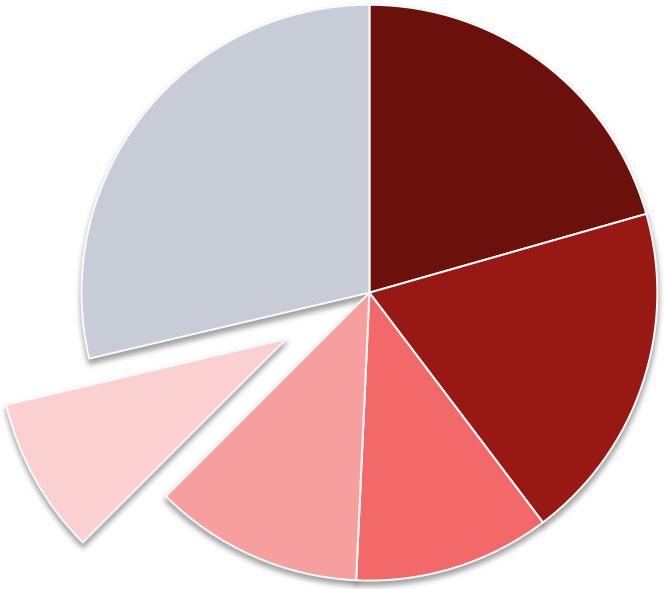


0-day Vulnerability Case Study

- Vulnerability details will be disclosed during the talk at the DEF CON conference
- Expected to patch before the conference
- If it is not patched, we will release the details publically in accordance with the Zero Day Initiative Vulnerability Disclosure Policy

Injections

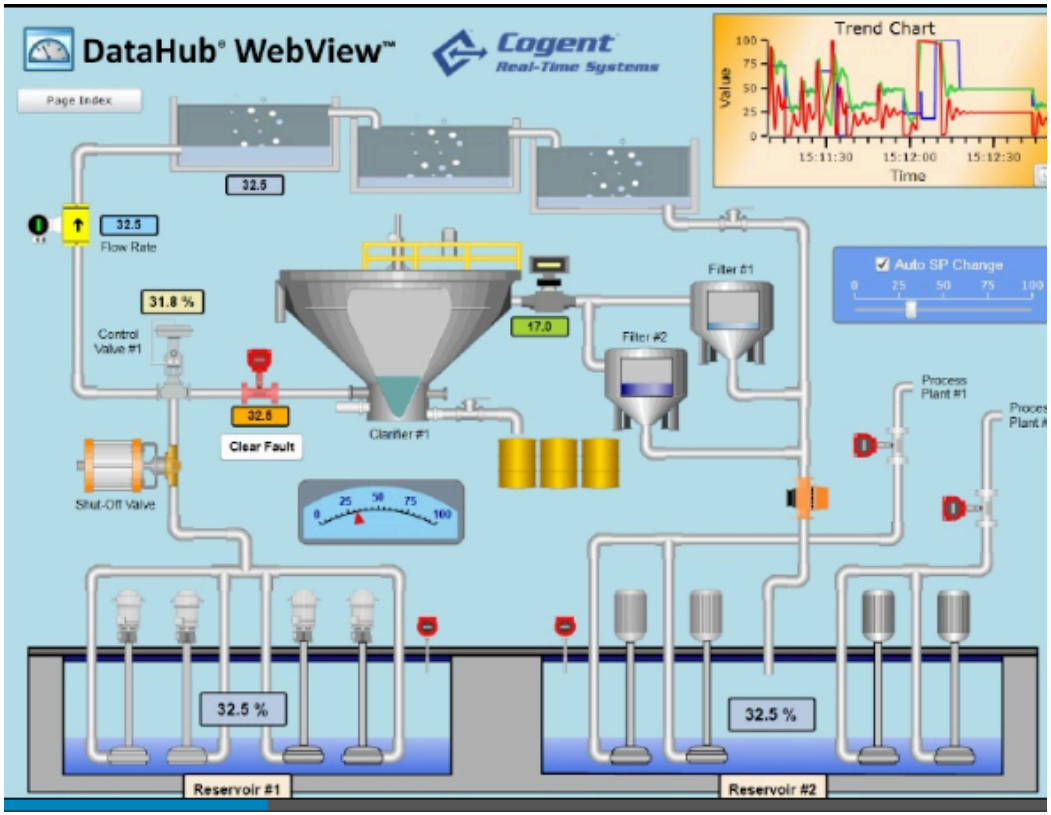
- 9% of identified vulnerabilities
- Common vulnerability types
  - SQL Injection
  - Code Injection
  - OS Command Injection
  - Command Injection
- Zero Day Initiative case study
  - Cogent DataHub Gamma Command Injection Remote Code Execution Vulnerability



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 EvalExpresssion 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

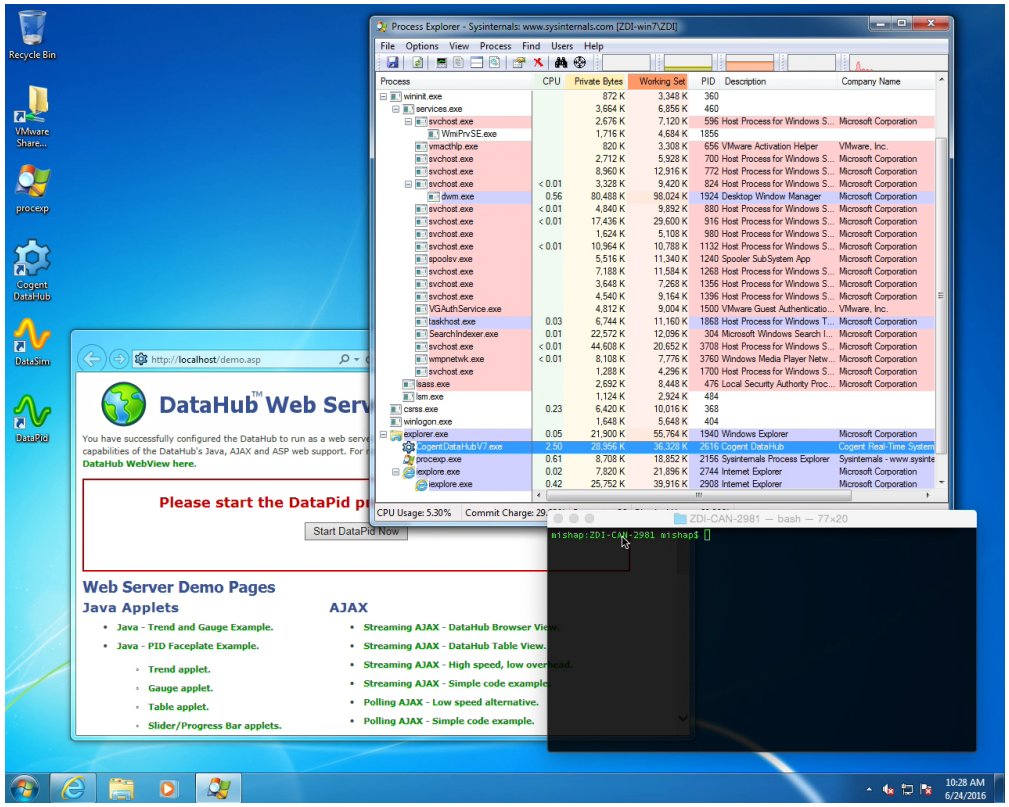
Vulnerable Code

```
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

Exploitation Demo



Patch Analysis

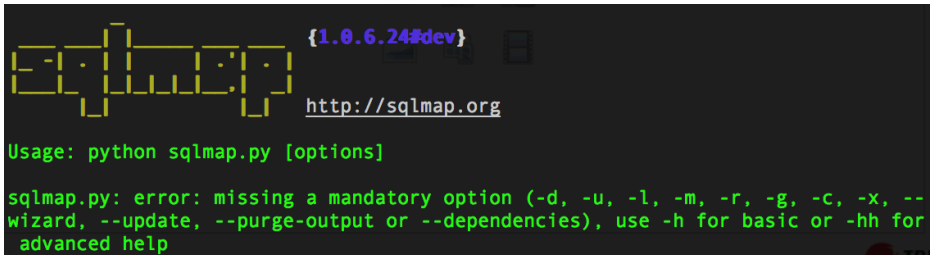
```
/* Any code to be run when the program gets shut down. */
method AJAXSupport.destructor ()
{
}

method AJAXSupport.AllowExpressions(enable)
{
        .allow_any_expression = (enable != 0 && enable != nil);
}

method AJAXSupport.XMLEscape (str)
{
        local           remainder = str;
        local           spot;
        str = "";
        while ((spot = strchr(remainder, '&')) != -1)
        {
                str = string (str, substr(remainder, 0, spot), "&amp;");
                remainder = substr (remainder, spot+1, -1);
        }
        str = string (str, remainder);
        remainder = str;
        str = "";
        while ((spot = strchr(remainder, '"')) != -1)
        {
                str = string (str, substr(remainder, 0, spot), "&quot;");
                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=\"0\" args=\"",
                        .XMLEscape(stringc(args)), "\"/></test>");
}
```

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

method AJAXSupport.XMLEscape (str)
{
        local           remainder = str;
        local           spot;
        str = "";
        while ((spot = strchr(remainder, '&')) != -1)
        {
                str = string (str, substr(remainder, 0, spot), "&amp;");
                remainder = substr (remainder, spot+1, -1);
        }
        str = string (str, remainder);
        remainder = str;
        str = "";
        while ((spot = strchr(remainder, '"')) != -1)
        {
                str = string (str, substr(remainder, 0, spot), "&quot;");
                remainder = substr (remainder, spot+1, -1);
        }
        str = string (str, remainder);
}
/* This method is dangerous.  It could allow somebody to execute arbitrary
   code via an HTTP call.  If you absolutely need it then create a script
   to define it, and then be sure the web server port is only accessible
   from a trusted network. */
/*
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=\"0\" args=\"",
                        .XMLEscape(stringc(args)), "\"/></test>");
}
```

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

Network Information

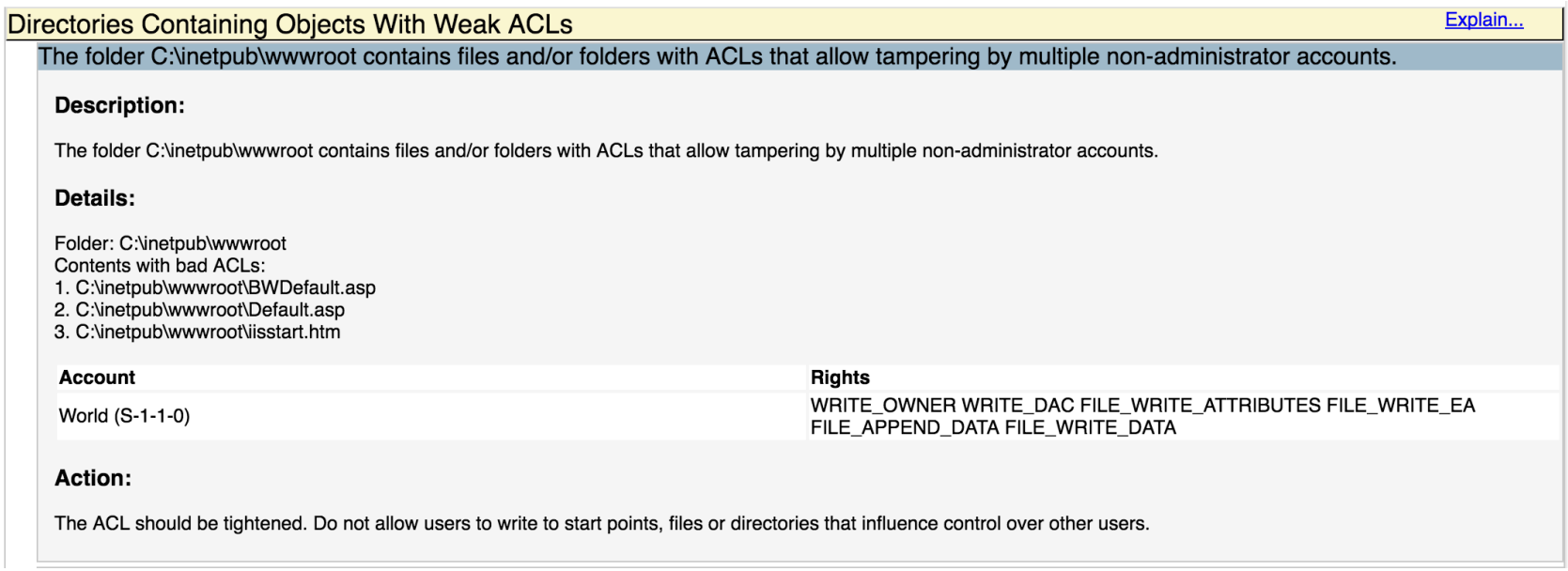
Ports

Explain...

Type	TCP	UDP
All New Ports (37 total)	12	2
Running as System	0	0
Running as Local Service	0	0
Running as Network Service	0	0
Running as Other	12	2

Port Name	State	Process	Account
5355/UDP -- Unknown Protocol	Unknown	svchost.exe (992)	
5355/UDP -- Unknown Protocol	Unknown	svchost.exe (992)	
4592/TCP -- Unknown Protocol	Listen	webvrpcs.exe (1992)	WIN-PC1TQDI11CB\ZDI
4592/TCP -- Unknown Protocol	Established	webvrpcs.exe (1992)	WIN-PC1TQDI11CB\ZDI

Attack Surface Analyzer Report



Directories Containing Objects With Weak ACLs

Explain...

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

Account	Rights
World (S-1-1-0)	WRITE_OWNER WRITE_DAC FILE_WRITE_ATTRIBUTES FILE_WRITE_EA FILE_APPEND_DATA FILE_WRITE_DATA

Action:

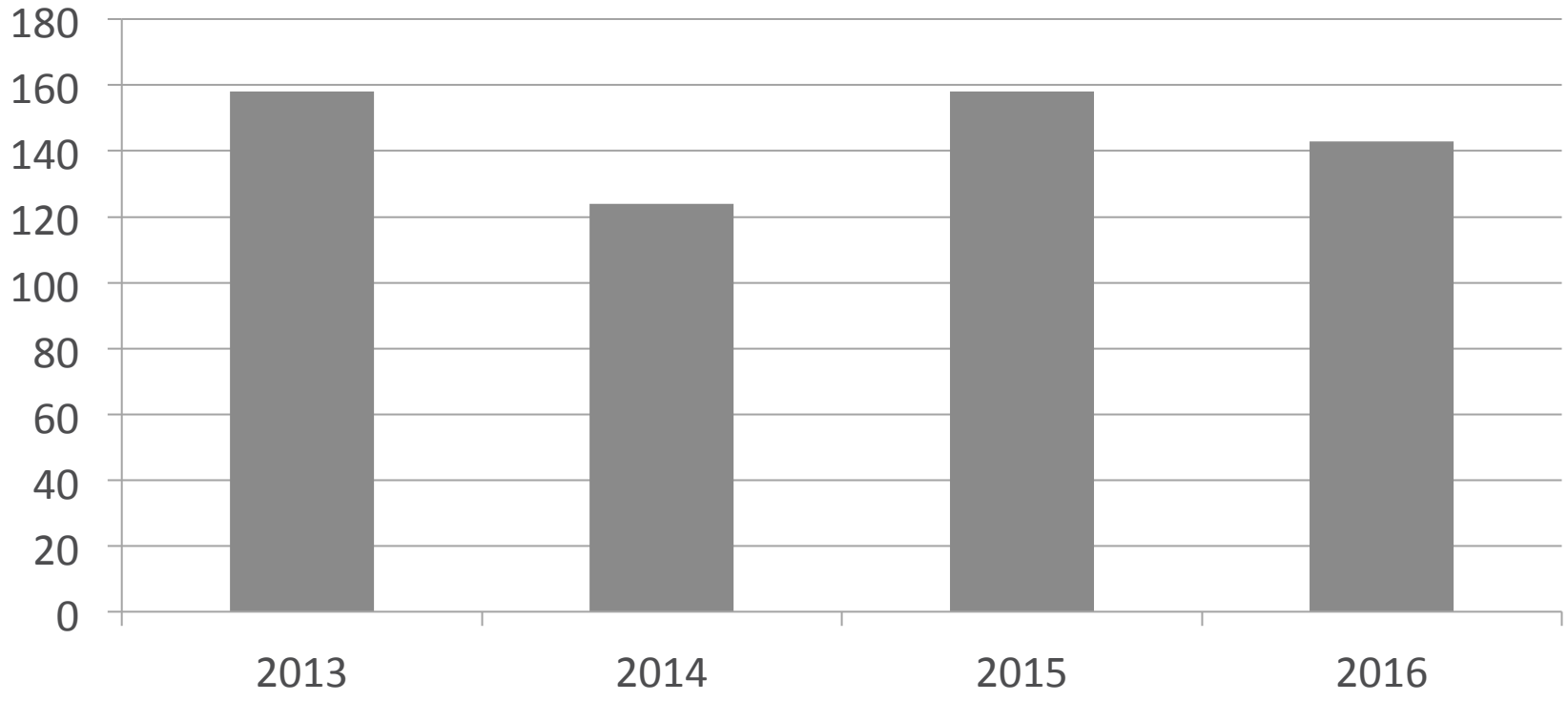
The ACL should be tightened. Do not allow users to write to start points, files or directories that influence control over other users.

Audit for Banned APIs

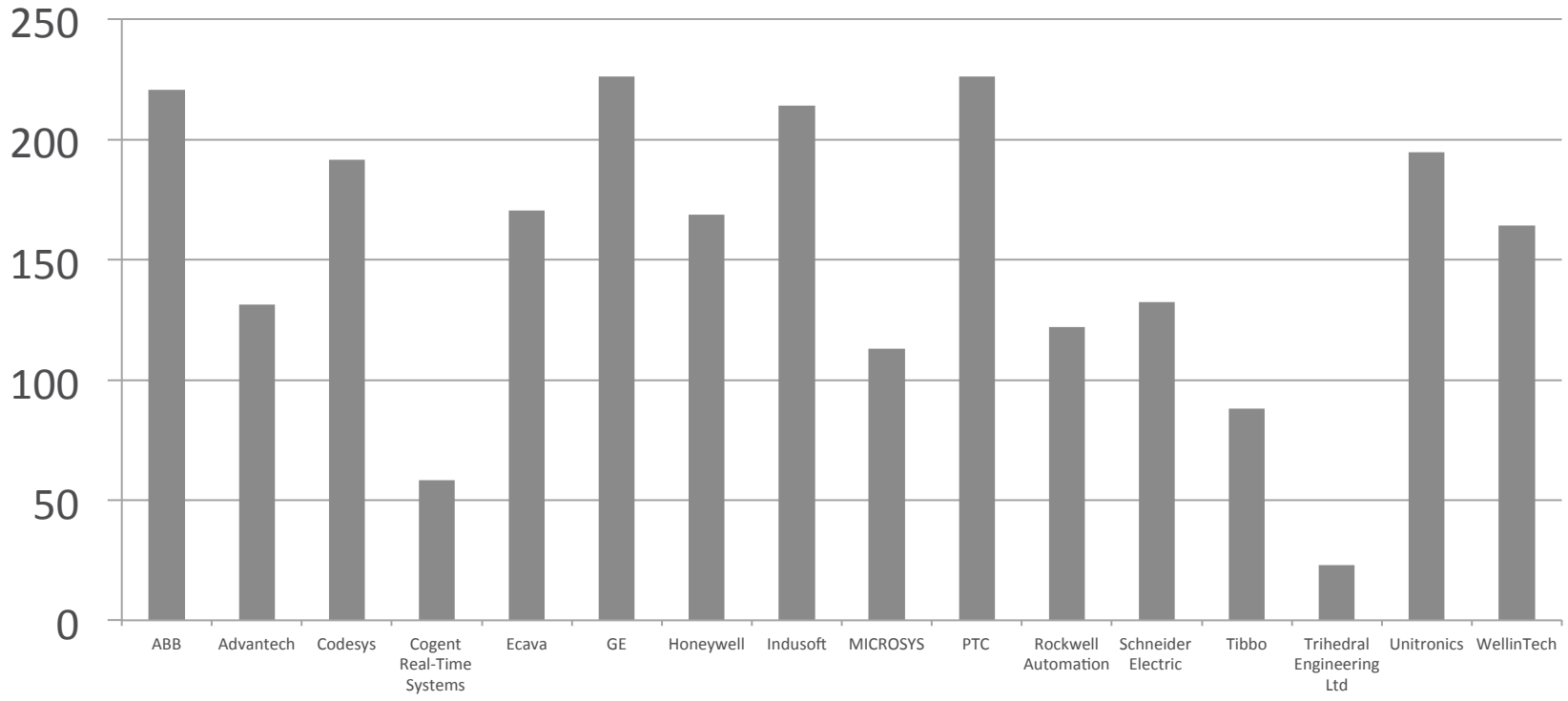
- C runtime has many APIs with serious security programs
- Microsoft banned use of problematic C library functions
  - "The Security Development Lifecycle" (Microsoft, 2006)
  - Security Development Lifecycle Banned Function Calls https://msdn.microsoft.com/en-us/library/bb288454.aspx
- Depressingly common in HMI code, with predictable negative impacts
- IDA is extremely valuable tool for auditing for inappropriate uses

Disclosure Statistics

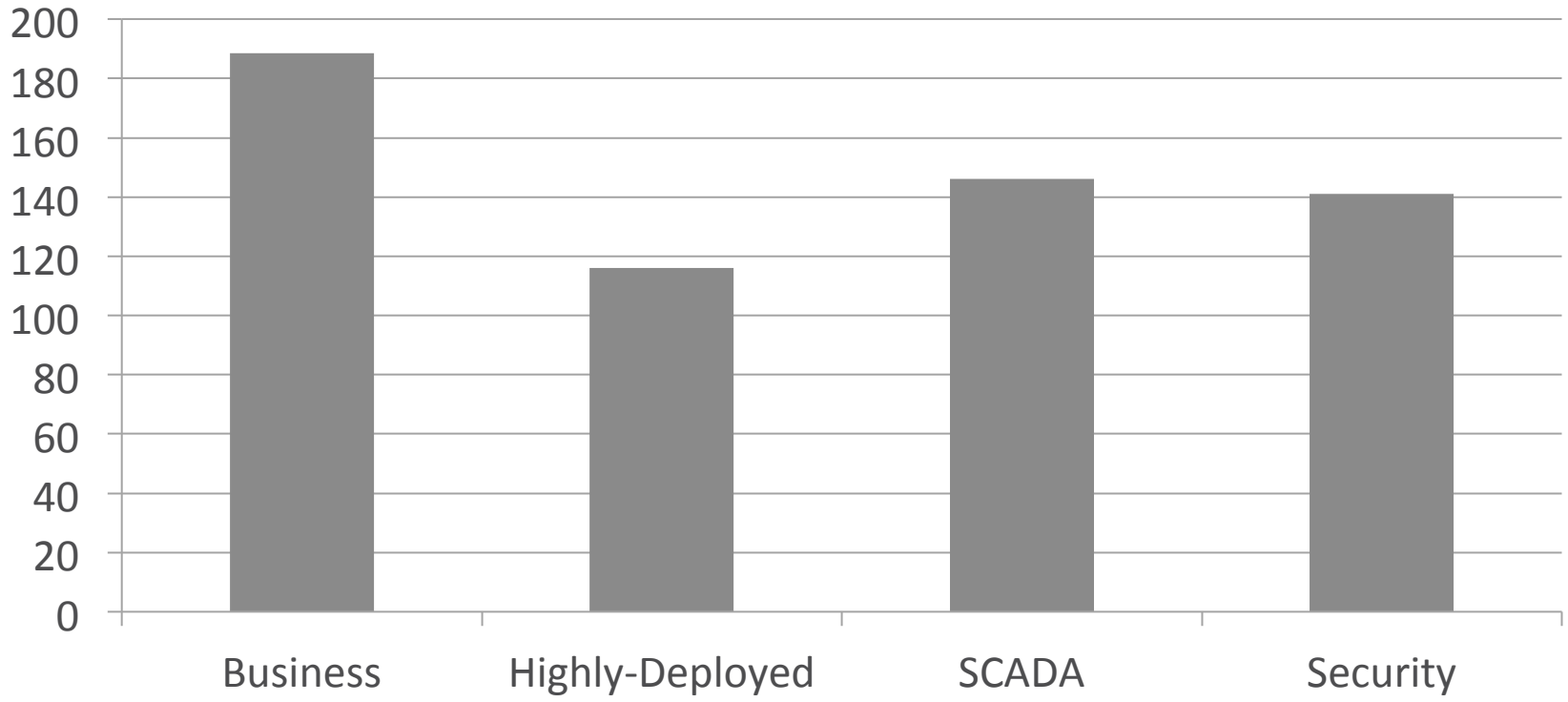
Vulnerability Exposure Windows



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

TREND MICRO

Questions?

ZERO DAY INITIATIVE

www.zerodayinitiative.com

@thezdi

TREND MICRO