

Aggregating and Correlating Security Events in Historians to Detect Cyber Attacks and Classify Attack Consequences

Kevin Lackey, Charles Perine, Dale Peterson
Digital Bond, Inc.
Fort Lauderdale, Florida
{lackey, perine, peterson}@digitalbond.com

Abstract: Timely detection of cyber attacks can limit or prevent adverse affects to a control system. Security events that can help detect attacks are located throughout the system in computers, field devices, network infrastructure equipment, security devices and other sources. Information Technology (IT) security vendors have developed Security Event Manager products that aggregate and correlate security events from a variety of sources to detect attacks. However these products lack interfaces to aggregate security events from control system applications and devices, and the Security Event Manager products do not exist today on most control system networks.

In this paper, we describe the how a control system Historian can be used to aggregate and correlate security events from both traditional IT sources and control system sources. An Event Taxonomy is described that is based on a composite approach. The aggregation and correlation for the Event Taxonomy is then implemented in a popular Historian, OSIsoft's PI server.

This project is based on Digital Bond's Portaledge research project.¹

Keywords: Historian, Security Event Manager (SEM), Attack Correlation, Security Event Aggregation, PI Server, ACE Module

1 Introduction

While the goal of a security program is to implement technical and administrative controls to prevent all attacks from affecting the confidentiality, integrity and availability of the system, it is likely this goal cannot be met. In addition, it can be helpful to know when an attack is taking place even when it is unsuccessful to identify and stop an attacker before they find a way to breach the defenses. Therefore attack detection is an important part of a cyber security program.

Cyber security information in the form of security events is available from a wide range of sources on a network. Operating systems, applications, network infrastructure,

¹ The Portaledge project discussed in this paper was funded by a research contract from the U.S. Department of Energy, National Energy Technology Laboratory.

security and other appliances, management systems and other devices on the network typically generate security related log events. Security events from any one of these devices are helpful in identifying attacks. However, aggregating and correlating security events from a diverse range of data sources is even better. This aggregation and correlation, if done correctly, can reduce the number of false positives and better determine the impact of an attack.

In the IT world there is a product class called a Security Event Manager² (SEM) whose purpose is to aggregate and correlate security data. The aggregation is for forensic and regulatory purposes, while the correlation is primarily for attack detection. A consortium of DHS and major oil companies funded Project LOGIIC [1], which used a SEM from ArcSight to aggregate and correlate security events and then detect cyber attacks on a simulated control system.

The detailed results on the security events aggregated and the correlation rules from Project LOGIIC have not been made public, but it is reasonable to assume that a full featured SEM used in the corporate IT world could be adapted to aggregate and correlate control system events. However the authors believe a SEM commonly used in the IT world for control system attack detection has at least two problems that will limit its acceptance and use in the control system community.

1. **There are not interfaces to the control system applications and components available from the SEM vendor.** A SEM vendor could create these interfaces, but given the relative small size of the control system market as compared to health care, financial, and other markets this is unlikely to happen unless funded by a control system asset owner. Additionally there is a wide diversity of control system components that would need interfaces and many are old, legacy devices and applications. At best, it would be years before significant coverage of required control system interfaces was available from an IT SEM vendor.
2. **The SEM does not exist on the control system.** Asset owners are hesitant to deploy any new device or application on the control system network. In addition, a SEM solution can cost more than \$100,000 and is a complex application that will have ongoing management and maintenance costs to be effective. It is orders of magnitude more complex than deploying a firewall or network intrusion detection sensor.

There is a control system application that aggregates and correlates events, has interfaces to a wide variety of control system applications and devices, and already exists in most control systems. It is a Historian.

Digital Bond has an ongoing research project, funded by the U.S. Department of Energy, to leverage a Historian's aggregation and correlation capability to detect cyber security attacks. This research project is named Portaledge.

² These products are also called Security Event Monitors or Security Incident Managers.

The Historian selected for use in Portaledge and discussed in this paper is OSIssoft's PI Server. The PI Server was selected because it has the majority market share in the oil/gas and electric sectors and because of its large number of control system and IT interfaces. The hundreds of available interfaces allows almost any control system and IT security events to be aggregated into the PI Server without writing any new interface code.

This paper discusses the methods used to aggregate and correlate security events in the PI Server [2] to detect cyber attacks. While the details and screen shots are specific to the PI Server, the correlation rules and event detail can be used by any Historian with an aggregation and correlation capability.

2 Expert System Approach

The initial approach to the Portaledge project was an expert system approach. The team generated attacks on control system applications and devices in Digital Bond's lab, and then the team identified the resulting security events that were available in log files. This included inspecting log files of firewalls, switch and net flow data, operating system logs, application logs, field device logs and more.

The attacks were generated with two different methods:

1. Attack Taxonomies – We used the MITRE Common Attack Pattern Enumeration and Classification (CAPEC) [3] attack taxonomy and began running through the attacks in each branch of the classification tree.
2. Expert Attack – Members of Digital Bond's offensive security team launched attacks from various points in the network, such as the corporate network, control center LAN and DMZ.

After these attacks, the potential security event logs were inspected for evidence of the attack. The team was then able to develop correlation rules that included the type, frequency and order of security events to detect the generated attacks.

This expert system approach was effective at identifying the attacks generated on the lab network. However the correlation rules tended to be very specific. If another expert deviated slightly in the order or technique, there was a possibility that the correlation rule would not detect the attack - - a false negative.

In addition to the false negative risk, we analyzed the percentage of the potential attacks that would be identified with this method. This method was not providing the coverage we desired so the team moved to a composite approach discussed in the following section. That said, it might be interesting to rerun the attack methods at the end of the project and see how the new composite approach does at identifying these attacks and what type of detail is available in the attack identification. The team hypothesizes that the attacks would be identified with the composite approach, but they would be less accurately classified than in the expert system approach.

3 Composite Approach

The second and current approach for correlating events in the Portaledge project is called a composite approach because there is a building hierarchy described in this section and pictured in Figure 1.

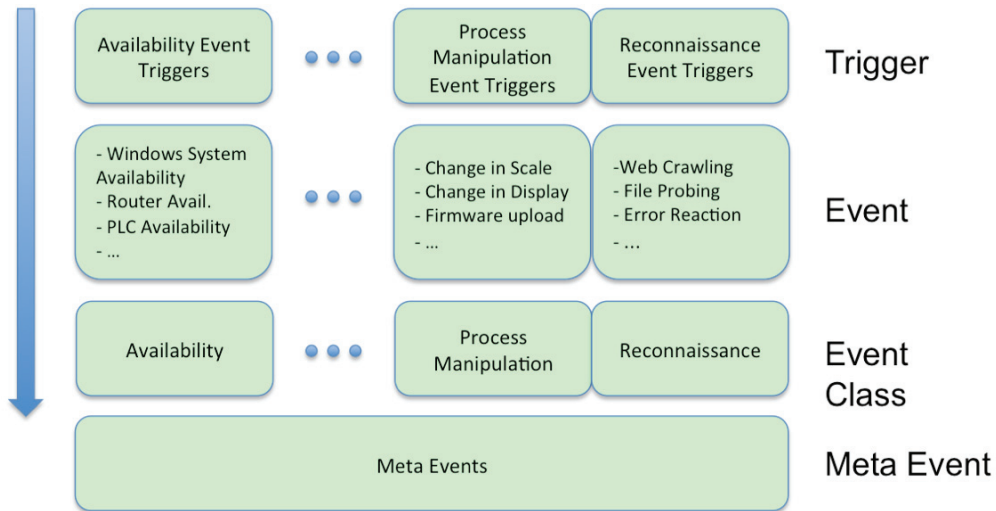


Figure 1 – Portaledge Event Taxonomy

3.1 Triggers

Security log entries, flow information, IDS alerts or other information is aggregated from a variety of data sources in the PI Server. This is done using one of the PI interfaces that is used today by asset owners to collect process data.

This data will be analyzed and may trigger an Event if it meets the Event criteria. An Event criteria could be a specific log message or a value, or values, in a specific log message, such as a value that exceeds a threshold.

Each Portaledge Event will have one or more triggers that are written into an Advanced Computing Engine (ACE) module for the Event.

3.2 Event

An Event is the lowest level piece of potential attack identification in the Portaledge project. Examples of Events are a Computer System Availability Event, an Unauthorized Control Event and New Administrator User Event.

Most events will have multiple triggers. For example the Computer System Availability Event has CPU Utilization, Memory Utilization, Hard Disk Space, Network Bandwidth and Network Latency triggers. The Ping, SNMP, TCP Response and Windows

Performance Monitor PI Interfaces are used to get this information into the PI Server for analysis by the ACE module and potential triggering of this Event.

The trigger or chain of triggers is also recorded with the Event. This is likely to be useful in real time or after incident analysis. The authors also believe that the length and variety of the chain could be part of a measure of confidence in the Event.

3.3 Event Classes

Each Event belongs to one and only one Event Class. The Event Classes planned for Portaledge as of January 2009 are:

- Availability
- Communication
- Enumeration
- Escalation
- Exploitation
- Obfuscation
- Process Manipulation
- Reconnaissance

When a single Event in an Event Class is generated, an Event Class Event is also generated. However each Event Class Event has a commonality, most often a time window that begins when the first Event in the Event Class triggers the Event Class Event. Subsequent Events in the Event Class during that time window, or other commonality, do not result in another Event Class Event, but each Event is added to the chain for the Event Class Event. This may be better understood by an example.

Imagine a malicious denial of service attack on a control system network. This would likely trigger a variety of different Availability Events, such as Performance Degradation, Windows Systems Availability, Network Device Availability and Simple Network Availability Events. If it affects all Windows systems, a Window System Availability Event may be generated for each Windows System. All of these Events during the time window are included in one Event Class Event and listed in the chain.

The length of the Event Class Event chain and variance of data sources and Event types in this chain are likely to be factors that will be used to determine the confidence or severity of the Event Class Event.

Many asset owners will choose to output Event Class Events and corresponding chains to a display in the control center. A security professional or trained operator will be able to view the chains and determine how widespread the problem is and perform whatever escalation or further analysis is required.

3.4 Meta Events

A Portledge Meta Event is created when two different Event Class Events occur with a commonality within a time window. The commonality could be a common data source. For example a server could be under attack and generate Enumeration, Exploitation, Escalation and Process Manipulation Event Class Events. The result would be a Meta Event with a chain of Event Class Events. There is only one Meta Event type and all of the variance and detail occurs in the related chain.

4 Security Event Aggregation in the PI Server

As the name indicates, the primary purpose of a Historian in a control system is to collect and store historical data. Many SCADA and DCS applications have a Historian component that is passed current process state data from real time control applications. Typically these Historians only gather data local to the Real Time Server.

There is another class of Historian, often made by a third party, for extending historian services across levels 2, 3 and 4 of the ISA99 standard reference model. Typically, these higher level historians include functionality to aggregate data from a variety of vendors and technologies. OSIsoft's PI server is one of these Historians that targets data collection and archiving from any source.

4.1 PI Interfaces

PI Interfaces communicate between data sources and PI Servers. Raw data from devices, such as field devices, computers and network equipment, feed into the PI Server via PI Interfaces. PI Interfaces can also send data from PI Servers to other systems, relational databases for example.

A PI Interface may communicate with a data source natively or use a common communication protocol. For control system data sources, the most widely used PI Interface is the OPC interface. While there are a variety of control system proprietary and standard protocols, most Windows based control system vendors support OPC. It has become the universal translator of control system protocols. There are also a large number of control system protocol interfaces, and it is difficult to find a control system device that cannot communicate with a PI Interface.

In addition to the control system specific PI Interfaces, there are a number of PI Interfaces that can collect data from traditional IT data sources such as firewall and router logs, IDS sensors, operating system logs, flow data, SNMP and more. Most of these IT interfaces are included in the IT Monitor interface package. The IT Monitor interfaces allow Portledge to aggregate both IT and control system data in the PI Server. Some of the more useful IT related PI Interfaces are discussed below.

4.1.1 Syslog

A syslog server receives and stores syslog messages that represent log events. Generic syslog messages contain the following three sections: PRI (Facility and Severity), Header, Message. The facility denotes the section of the operating system that created the

message, e.g. kernel message, user-level message and security message. The severity is set at the syslog event source at a level from 0:Emergency to 7:Debug, which determines what messages are sent to the syslog server. The header contains the date/time of the event and hostname of the system that created the message. The message provides the content of the log message.

The Syslog Server PI Interface can be used to log all system event messages on systems that support syslog. This is an incredibly valuable source of information for detecting cyber attacks and other security incidents. Once the syslog messages are in the PI Server they can be analyzed to identify Portaledge Events

There are a large number, probably the vast majority, of syslog messages that are unrelated to security. Filtering can be turned on so only security events are sent to PI.

4.1.2 IPflow

Most of the popular routers and switches used in control systems provide NetFlow, sFlow or IPFIX data that characterizes the data flowing through the router or switch. This data is sent to a collection point where an administrator can view information about the network. The following information is typically collected: Source Address, Source Port, Destination Address, Destination Port, IP Protocol, and Number of Bytes. This data can then be parsed to discover systems that are performing network scans, both host discovery and port scans, worm propagation, new systems on the network and increased network traffic.

4.1.3 Windows Performance Monitor

The Windows NT 4.0, Windows 2000, Windows XP, Windows 2003, Windows Vista and Windows 2008 operating systems contain the Windows Performance Monitor (PerfMon). This tool provides detailed system information to administrators and users. PerfMon can be setup to graphically represent the use and performance of a system. Data can be collected from numerous system sources including the following: processors, memory, hard drives, network devices, jobs, swap space. Using these data points, an administrator can assess the overall health of a Windows workstation or server. Windows PerfMon objects are extensible and often available to monitor layered applications (eg. AntiVirus statistics).

The PerfMon PI Interface can be used to log system usage statistics. An increase in network traffic, processor or memory usage may be indicative of increased activity on the control network, an attack or malware.

4.1.4 SNMP

The Simple Network Management Protocol (SNMP) is used to monitor network attached devices. The SNMP PI Interface can collect a wide range of data from a number of systems. Some field devices provide SNMP data, as well as firewalls, routers and computers. CPU usage, memory usage, network statistics, disk status and firewall statistics are a few examples of data that can be collected via SNMP. The information collected by the SNMP interface depends on the information the device provides. The

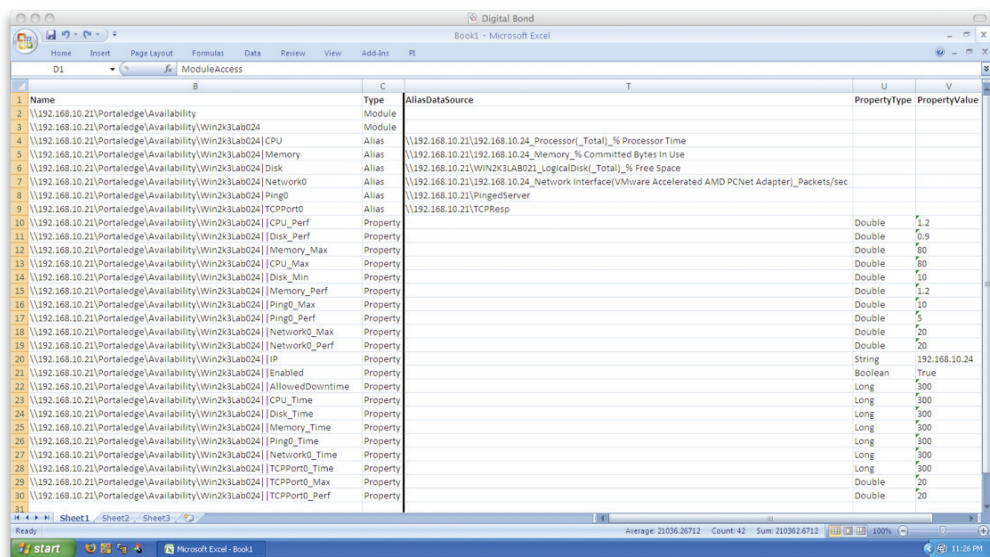
data collected by the SNMP interface can provide valuable information regarding the status of a device or indications of an attack much the same way the Windows PerfMon PI Interface can.

4.2 PI Tags and PI Values

A PI Tag represents archive storage pointers for time series values and associated configuration attributes for access, collection, storage and display controls. A tag must be created for each data item collected by a PI Interface. For example, a tag can be the CPU usage of a single computer while another tag would be the memory usage for the same machine. The PI Interfaces [4] have detailed documentation on how to setup tags and what type of data each PI Interface can provide to the tags.

Values are collected and recorded for each PI Tag. A PI Value has a timestamp, value and other attributes (eg. Annotations). A tag that is setup to ping a system every minute will create a new PI Value every minute when it receives the result of the ping. Interfaces test PI Values for significant changes which are then recorded by PI Servers. PI values can be analyzed by a human, ACE or another application.

One of the more painful tasks in any Historian, including the PI Server, is adding all of the tags that are required. The typical number of tags for the Availability Event Class is six tags per machine. A small network may have 20 machines; meaning the PI administrator would have to create 120 tags with the system management tool user interface. Fortunately, OSIsoft provides a way to make this process easier. Using Excel and the PI System Management Tools, a user can create a number of tags in a very short amount of time. Digital Bond will include in the Portledge release package an Excel template to aid in the creation of the required tags. Once the PI Interfaces are setup, it is relatively simple to create the tags for 20 or even 100 or more machines.



	Name	Type	AliasDataSource	PropertyType	PropertyValue
1	\\192.168.10.21\Portledge\Availability	Module			
2	\\192.168.10.21\Portledge\Availability\Win2K3Lab024	Module			
3	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\CPU	Alias	\\192.168.10.21\192.168.10.24_Processor\Total\% Processor Time		
4	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Memory	Alias	\\192.168.10.21\192.168.10.24_Memory\% Committed Bytes in Use		
5	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Disk	Alias	\\192.168.10.21\Win2K3Lab021_LogicalDisk\Total\% Free Space		
6	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Network0	Alias	\\192.168.10.21\192.168.10.24_Network Interface\VMware Accelerated AMD PCNet Adapter\Packets/sec		
7	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Ping0	Alias	\\192.168.10.21\PingedServer		
8	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\TCPPort0	Alias	\\192.168.10.21\TCPResp		
9	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\CPU_Perf	Property		Double	1.2
10	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Disk_Perf	Property		Double	0.9
11	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Memory_Max	Property		Double	80
12	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\CPU_Max	Property		Double	80
13	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Disk_Min	Property		Double	10
14	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Memory_Perf	Property		Double	1.2
15	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Ping0_Max	Property		Double	10
16	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Ping0_Perf	Property		Double	5
17	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Network0_Max	Property		Double	20
18	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Network0_Perf	Property		Double	20
19	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\IP	Property		String	192.168.10.24
20	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Enabled	Property		Boolean	True
21	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\AllowedDowntime	Property		Long	500
22	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\CPU_Time	Property		Long	500
23	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Disk_Time	Property		Long	500
24	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Memory_Time	Property		Long	500
25	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Ping0_Time	Property		Long	500
26	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\Network0_Time	Property		Long	500
27	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\TCPPort0_Time	Property		Long	300
28	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\TCPPort0_Max	Property		Double	20
29	\\192.168.10.21\Portledge\Availability\Win2K3Lab024\TCPPort0_Perf	Property		Double	20

Figure 2 – Creating Tags Using PI System Management Tools

4.3 Module Database

The correlation rules in ACE modules will expect the tags to have a specific naming format. This conflicts with the usual practice of each organization having their tag naming practice. Therefore there is an intermediate step required to convert the asset owner tag names to alias tag names used in the ACE modules. This is done in the PI Server's Module Database [5].

The Module Database allows for a flexible naming convention for control system operator tags and allows the ACE Module to easily scan through all of the aliases. Along with providing a template to create the tags, Digital Bond will be providing a template to create the alias for the Module Database.

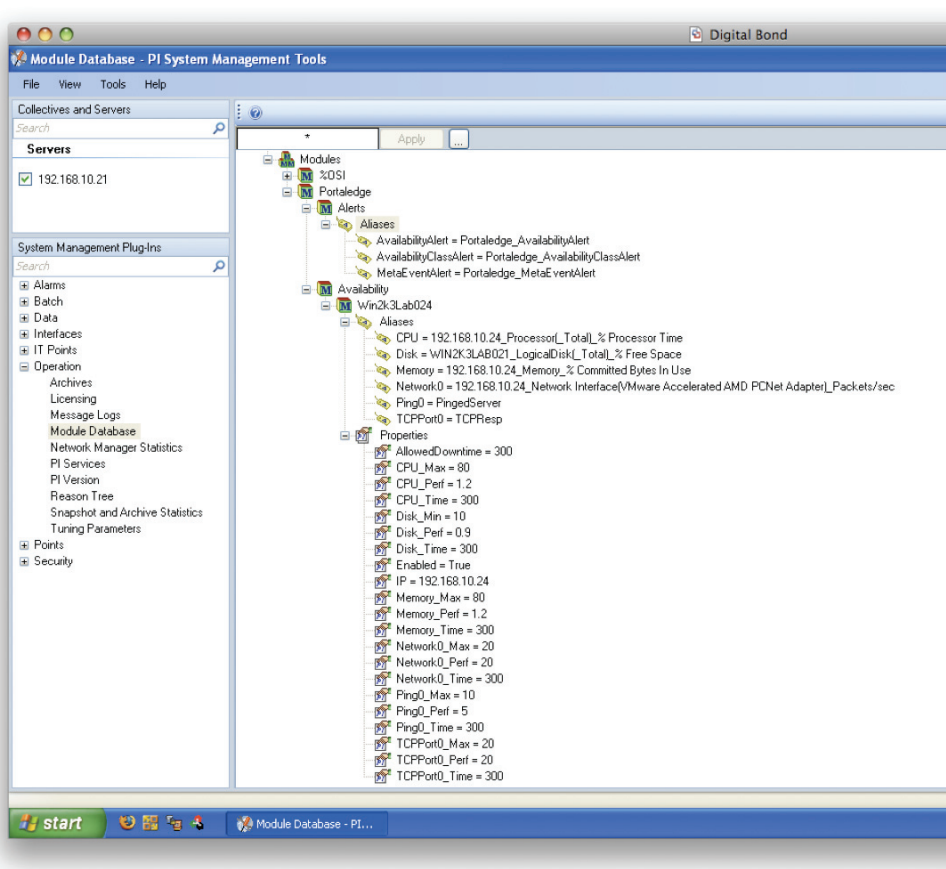


Figure 3 – Creating Alias with the Module Database

A tag can have more than one alias. In Portaledge, an alias is created for each instance of a tag in an Event ACE Module. So if a single ping tag is used in the three ACE Modules, three aliases are created. This is not necessary, but was done for clarity and performance reasons.

5 Correlation in the PI Server

The PI Server in its role as a Historian has an Advanced Correlation Engine (ACE) [6] that is used to calculate key performance indicators, preventive maintenance information, provide billing information and other analytics using control system data.

The ACE wizard is a plug-in to the MS Visual Studio IDE. The plug-in allows a programmer to retrieve data stored in the PI tags and work with that data in Visual Basic. Programs are built as ACE modules. Input variables and output results are abstracted through aliases instead of hardcoded PI tags. This approach enables portability and reuse of Portaledge analytics without additional coding. ACE module results are usually saved in PI but can be exported to files or other applications.

From a Portaledge perspective, ACE provides the team with an almost limitless and simple ability to implement correlation rules. Once the security events have been aggregated into the PI Server and mapped to the alias that will be universal in Portaledge ACE modules, any correlation method that can be thought of can usually be programmed as an ACE module.

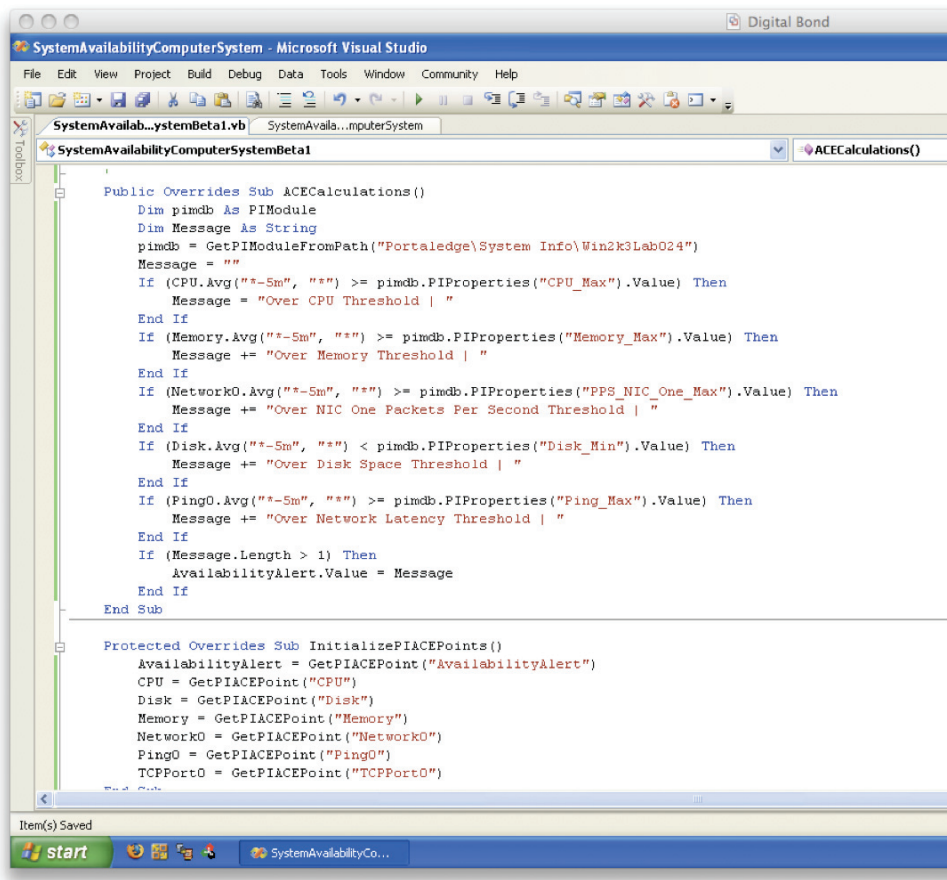


Figure 4 – System Availability Event ACE Module

Each defined Portaledge Event has a corresponding ACE Module; each Event Class Event has a corresponding ACE Module; and there is one ACE Module for a Meta Event. Control system operators will be able to select and deploy the ACE Modules that make sense for their system. For example, Digital Bond expects most Portaledge deployments to include many if not all of the Availability Events. However, the system operator may choose to deploy Exploitation and Process Manipulation Events and not to deploy Enumeration Events that have a less serious impact on their own.

Control system operators will have limitations based on the data sources available in the SCADA or DCS. While the PI Server can aggregate data from almost any data source, by default it does not collect or generate security events. Monitoring interfaces and Portaledge ACE Modules are essential for generation of security events.

The beta package will include a set of ACE modules for Availability Events. One of these modules will check to see if user-defined thresholds have been exceeded. The module will determine if the CPU, memory, disk space, network traffic, ping or other network ports on the system have passed or fallen below a specific boundary and will create an alert. Should the disk space fill and the network traffic increase this could be an indication that file sharing is occurring on this particular system while an increase of network traffic, CPU and memory could be an indication of a worm.

The thresholds and time windows in the ACE Modules can be customized by a Control System Operator. Digital Bond is selecting what we believe are reasonable default thresholds and time windows, but the release package will also include documentation on how to configure these settings.

6 Display Options

There are a number of different ways to display the output from the ACE Modules. By default, the output of an ACE Module is sent to a tag. Since ACE Modules are written in Visual Basic, the output can also be sent to a file, a database or to another system such as email, syslog, a SEM or some other logging application.

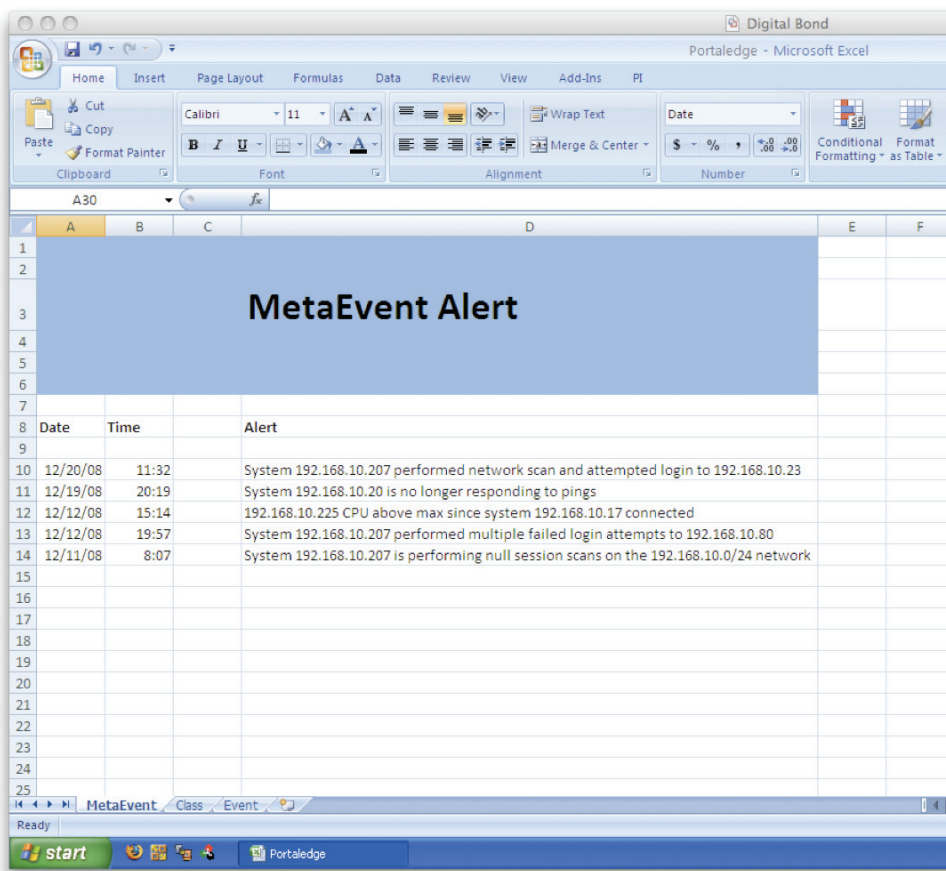
Displays are typically highly customized for each control system. What one system operator prefers is often unacceptable to another system operator. Digital Bond envisions a great deal of customization of the Portaledge Event, Event Class Events and Meta Events, along with their corresponding chains, in displays.

One approach is to display the Event Class Events along with their chains on display, and then have a programmatic means to click through the Events in the Event Class Event chain to display additional detail. The authors can also envision trend charts showing the number of Events. While control system operators are not typically cyber security professionals, it would be possible to have them monitor displays and have rules for escalation to cyber security professionals. Given the isolated nature of control systems, this should be a rare occurrence rather than resulting in nightly or hourly pages.

At least one proof of concept display will be provided in the Portaledge package. This proof of concept display will be based on an Excel spreadsheet and uses OSISoft's

Datalink plug-in. The display will show three different levels of data coming from the ACE modules.

1. The Meta Event Class data will be displayed on the main screen and will contain information the overall status of systems on the network, provide information on chained events as well as singularities.
2. Event Class data will be displayed on another page allowing the user to dig deeper into alerts given on the Meta Event Page.
3. Finally a third page will contain the individual events, allowing the user to perform further investigations.



Date	Time	Alert
12/20/08	11:32	System 192.168.10.207 performed network scan and attempted login to 192.168.10.23
12/19/08	20:19	System 192.168.10.20 is no longer responding to pings
12/12/08	15:14	192.168.10.225 CPU above max since system 192.168.10.17 connected
12/12/08	19:57	System 192.168.10.207 performed multiple failed login attempts to 192.168.10.80
12/11/08	8:07	System 192.168.10.207 is performing null session scans on the 192.168.10.0/24 network

Figure 5 – Datalink Display of Meta Event

7 Generalized Solutions

The Portaledge project has been developed on OSISoft's PI Server and components. That said, it is a goal of the project to generalize the results so they can be used by other control system Historians that have an aggregation and correlation capability. Detailed documentation is available that defines the data sources, triggers, and tags/points. The Visual Basic code for the correlation rules that cause Events, Event Class Events and Meta Events is also available. This code should be easily convertible to a format acceptable to other Historian correlation engines.

8 Future Development

The most obvious area for future development is the creation of additional Events and Event Class Events. The current Portaledge Events and Event Classes are not meant to be exhaustive or complete. They were selected to try a variety of different scenarios as well as taking a "low hanging fruit" approach. The existing Events are also likely to undergo significant modification after practical use by the control system community.

The authors would also like to take attack taxonomies discussed in Section 2 and see how effective Portaledge is at detecting these attacks.

The current Portaledge approach focuses on alerting to detect security incidents and aggregating data for after incident investigation. One area the authors believe could be interesting and valuable is to use the aggregated data to calculate metrics on the security state of the control system at any point in time. This metrics or series of metrics could be displayed on an operator console as either a red, yellow, or green status light, as a gauge showing a security level between 0 and 100, or some other simple to interpret visual value. This would then fit into a control system operators typical day-to-day activity of watching for alerts and taking actions based on an alert.

About the Authors – Kevin Lackey is a Senior Security Researcher at Digital Bond, Inc. He is the project lead on the Portaledge project and has contributed on the Bandolier project. Prior to joining Digital Bond, Mr. Lackey was a Senior Scientist at Idaho National Lab where he performed penetration testing and vulnerability analysis on critical infrastructure control systems. Mr. Lackey has a BS in Computer Science from Idaho State University.

Charles Perine is a Security Researcher at Digital Bond, Inc. He is a PI Server lead on the Portaledge project and has contributed on the SCADA Honeynet and Quickdraw projects. Prior to joining Digital Bond, Mr Perine worked on control system security projects at Sandia National Labs. Mr. Perine received his B.S. in Computer Science from California State University Hayward.

Dale Peterson is a founder of Digital Bond, Inc. and has led their Control System Security Consulting and Research Practice since 2000. He began his career as an award winning Cryptanalyst at the National Security Agency and has more than twenty years experience in information security.

References

- [1] LOGIIC Cyber Security System,
<http://www.cyber.st.dhs.gov/docs/LOGIICbrochure.pdf>, 2006.
- [2] OSIsoft, Inc., PI Server
<http://www.osisoft.com/products/pi%20system/pi%20server.htm>.
- [3] MITRE, Common Attack Pattern Enumeration and Classification,
<http://capec.mitre.org/index.html>.
- [4] OSIsoft, Inc., PI Interfaces, <http://www.osisoft.com/products/pi+interfaces.htm>.
- [5] OSIsoft, Inc., Module Database,
<http://www.osisoft.com/products/pi+module+database.htm>.
- [6] OSIsoft, Inc., Advanced Computing Engine (ACE),
<http://www.osisoft.com/products/pi+advanced+computing+engine.htm>.