

Self-configuring deep protocol network whitelisting

Game-changing technology that boosts ICS security while reducing operational costs

Authors

Dr. Sandro Etalle, sandro.etallesecmatters.com
Dr. Clifford Gregory, cliff.gregorysecmatters.com
Dr. Damiano Bolzoni, damiano.bolzonisecmatters.com
Dr. Emmanuele Zambon, emmanuele.zambonsecmatters.com

15 September 2014

Version 2.1

Contents

1	The Critical Infrastructure Cyber Threat	2
1.1	Increase in Cyber Attacks	2
1.2	ICS Vulnerabilities	3
2	Traditional Detection Methods	5
2.1	Blacklisting and Signature-Based Detection	5
2.2	Anomaly Detection	6
2.3	Whitelisting	7
3	A Novel Detection Method	9
3.1	Deep Protocol Behavior Inspection (DBPI)	9
4	SilentDefense ICS	12
4.1	Additional Benefits of SilentDefense ICS	13
5	Case Studies	15
5.1	Threat #1: Chronic exposure to unknown software vulnerabilities	15
5.2	Threat #2: Combat Internal Theft	16
5.3	Threat #3: Downtime from malware	16
5.4	Threat #4: System configuration in production shifts from expected blueprint	17
6	Closing	19
6.1	Acknowledgments	19
6.2	About SecurityMatters	19
6.3	About the Authors	20
A	Understanding DPBI	22
A.1	The Learning Phase	22
A.2	The Tuning and Customization Phase	22
A.3	The Detection Phase	24
B	Frequently Asked Questions about DPBI	25

1 The Critical Infrastructure Cyber Threat

Cyber attacks against Critical Infrastructure (CI) – energy, gas, oil and water – are constantly growing. ICS-CERT published in its Incident Response Summary Report¹ that in 2009, eleven incidents were reported. By 2012, the number increased by about 2000% to 198. Water and energy organizations accounted for about half of all reported incidents.

The CI industry is subject to different threats than organizations in other sectors, as for instance financial institutions. Attacks against CI are mostly carried out by motivated and well-funded criminal organizations, competitors, cyber terrorists/activists and even state-sponsored government agencies. One example of a state-sponsored cyber attack, the Stuxnet worm, was used to slow down the Iranian nuclear program in 2010. As the first cyber weapon to gain widespread public attention, it is the tip of a rapidly growing iceberg that is impacting the entire CI industry.

The severity of this issue has prompted many countries, including USA and EU, to craft legislation that mandates protection of CI and SCADA systems. Such attention is necessary to sustain a quality of life as well as the reliability, productivity and profitability of CI in the face of increasingly high threat levels. Failure in this area represents a huge risk to economic and public safety.

1.1 Increase in Cyber Attacks

The increase in the number of cyber attacks is due to a mix of two primary factors:

1. Industrial Control Systems (ICS) were once made up of proprietary, non-standard hardware and software. Vendors could keep their clients' systems secure simply by not disclosing their specifications. Each system was custom built for a client and therefore no two systems were alike. Nobody but the manufacturer and the owner of the system knew how they

¹https://ics-cert.us-cert.gov/pdf/ICS-CERT_Incident_Response_Summary_Report_09_11.pdf

worked. Attackers had virtually no starting point to craft an attack and little advantage. Security through obscurity worked very well. More recently, for reasons of interoperability, efficiency and cost, vendors have begun employing standard off-the-shelf operating systems, software and hardware. The cost to acquire these systems has come down substantially for everyone, including attackers. Specifications for these systems are known, and since prices have come down, attackers simply purchase them for their research and quickly find vulnerabilities in many CI systems. Windows-based installations are but one example of these. This set the groundwork to allow determined attackers to get in. On top of this, since legacy process control devices were designed with the assumption that they would be accessed only by legitimate users, security was not part of their design. As a result, these devices often lack of even the most elementary security mechanisms.

2. ICS have historically been physically isolated from each other, even within the same organization. For instance: valves, monitors, and alarms for oil and gas producers were accessible only by a person on the actual site of the field devices. Techniques for securing physical locations have been well understood, tested and developed over a number of decades, resulting in a very high degree of security. In the last few years, however, organizations have begun connecting their systems within digital networks to take advantage of the enormous gains in efficiency offered by new technology. At that same time, however, the fluid and silent nature of digital networks make these field devices reachable to attackers from different locations. In 2011, researchers demonstrated that some critical devices were visible – and exploitable in real-time – with a simple web search.

Whereas the CI industry could once count on obscurity and lack of access to guarantee security, the recent technology shift provides an opportunity and the means for attackers to have the advantage over typical ICS. For high-value CI, there is plenty of motivation to attempt an attack. Now, with the old barriers essentially reduced to a speed bump, it is a perfect storm of an opportunity for cyber criminals to pursue.

1.2 ICS Vulnerabilities

Digital Bond, a security consulting firm, ran a PLC hacking experiment called Project Basecamp² in order to show how CI systems can be easily subverted. The team acquired a variety of legacy systems that are widely used in production today. Using off-the-shelf security tools, Digital Bond tested those systems for common and advanced vulnerabilities. The results (presented in the graph below) show that every tested system was susceptible to multiple, easy to exploit vulnerabilities.

²<http://www.digitalbond.com/tools/basecamp/>

	A-B Quality	Schneider Electric	General Electrics	Schweitzer Engineering Laboratories	Koyo
Firmware	!	X	!	!	!
Ladder Logic	!	!	X	!	X
Backdoor	!	X	X	✓	✓
Fuzzing	X	X	X	!	!
Web	!	X	N/A	N/A	X
Basic Config	!	!	X	!	!
Exhaustion	✓	✓	X	✓	✓
Undoc Features	!	X	X	!	!

Table 1.1: Key: **X**= vulnerability exists, easy to exploit; **!**= vulnerability exists, difficult to exploit; **✓** = no vulnerability found. Image Credit: Digital Bond

2 Traditional Detection Methods

There are three main methods to detect network-based attacks, which we review below: blacklisting (signature-based detection), anomaly detection, and whitelisting.

2.1 Blacklisting and Signature-Based Detection

Most existing security solutions are *signature-based* and rely on experts *blacklisting* known attack patterns. This method can only intercept and prevent the spread of previously analyzed threats and is easily circumvented by morphing attack payloads. Every time a new threat is discovered, a new signature to detect it must be developed and distributed; it can take weeks or even months before all systems are updated. In the meantime, malware strains that have not yet been isolated, analyzed and mapped to a signature can propagate without notice. For this reason, malware such as Stuxnet, Duqu, Flame and Shamoon, spread across systems silently for lengthy periods of time. Stuxnet leveraged four zero-day vulnerabilities and went undetected for at least a year.

Project Basecamp confirmed the limitations of signature-based solutions by disclosing dozens of previously unknown vulnerabilities for which no signature had yet been crafted. Zero-day vulnerabilities remain a gaping hole in the security of those functions like water, gas and electric utilities. What's worse, new exploits can be bought on an open and thriving Internet black market that grows daily, giving criminals a decisive advantage by increasing their knowledge base and reducing the amount of time to launch an attack.

At the present state of affairs, it is impossible to have a set of signatures that would provide a reasonable degree of protection to CI systems. Blacklisting will never provide sufficient coverage in the CI world, where attackers are more motivated to remain silent, and where attacks are targeted to specific systems. However, bearing in mind its limitations, blacklisting still makes sense to provide at least partial protection in the Internet domain because it is easy to implement and works well in its limited scope.

Alternatives to Blacklisting To counter the newest and most sophisticated cyber threats and state-sponsored attacks, and to protect adequately CI systems it is necessary to move

away from the universally accepted signature-based security paradigm. It is not feasible to isolate attacks one by one and extract a suitable signature for each of them.

Computer scientists have been working for more than a decade on techniques and algorithms to detect unknown cyber threats. The two main approaches are anomaly detection and whitelisting. In principle, anomaly detection and whitelisting share a common vision: to model the behavior of *legitimate* network traffic and flag when anomalies are detected, or block them altogether. In practice, they have evolved into two different approaches. Anomaly detection enjoys automatic configuration at the cost of a less accurate analysis and a higher false positive rate, while whitelisting requires manual configuration but can afford a much more fine-grained analysis.

2.2 Anomaly Detection

Most anomaly detection systems are based on artificial intelligence algorithms, like neural networks, that first *learn* what are the normal, legitimate “states” of the system and afterwards raise an alert when the presence of an anomaly is detected. These algorithms have been applied successfully in the past to other fields. For instance, closed circuit television systems use artificial intelligence for recognizing a suspect among groups of people passing through border security.

When applied to network monitoring, anomaly detection comes with two inherent flavors, depending on whether the analysis it makes is qualitative or quantitative.

Qualitative Anomaly Detection Intuitively speaking, *qualitative*¹ anomaly detection systems analyze network packets one by one and raise an alert when the content of one packet is “too different” from “the norm”. Despite some promising results seen in initial studies, serious shortcomings became obvious when these kinds of systems were deployed in real environments. When processing real network traffic, this approach falls short of identifying malicious behavior with sufficient accuracy [1].

Tweaking is essential in a dynamic environment such as a computer network. Because qualitative anomaly detection systems mostly work like black boxes, it is difficult, if not impossible, for users to tweak working parameters to improve detection. Additionally, most qualitative anomaly detection algorithms are based on the assumption that it is possible to seize network traffic’s intrinsic characteristics by using a generic approach. The most widely used such approach is N-gram analysis. N-grams are sequences of N- consecutive bytes extracted from network traffic data. During an initial training phase, N-gram analysis computes statistics about N-grams present in normal network traffic and builds traffic models. Later, packet payloads are compared to these models, and when the “distance” between current traffic and traffic models surpasses a given threshold, the traffic is flagged as anomalous and thus suspected malicious. The idea behind this technique is that payloads used in cyber attacks feature a different set of N-grams than regular traffic. Recent studies [1, 2] prove not only that this assumption is often wrong, but also that systems based on N-gram analysis generate an extreme percentage of false alerts even when processing regular attack-free traffic, making them too inaccurate and far too expensive to manage. N-gram analysis turned out to be too coarse a method to achieve effective analysis of network traffic data in order to detect malicious actions, mostly because it discards both syntax and semantics of the underlying data.

Quantitative Anomaly Detection There are very few commercial anomaly detection systems available, and they are mainly based on *quantitative* analysis of network flows. In theory, quantitative anomaly detection systems are also able to detect unknown threats. In practice,

¹In the scientific literature, qualitative anomaly detection systems are called “payload-based” anomaly detection systems.

those solutions can detect only threats that generate spikes in data volumes and network communications such as a denial of service attacks. This represents a severe limitation because most sophisticated attacks are aimed at stealing data or causing long-term damage, and they would remain undetected because they do not generate a spike behavior in network flows.

2.3 Whitelisting

Network whitelisting consists of identifying and detailing legitimate and acceptable network traffic, and blocking or alerting when non-matching traffic is detected. The effectiveness and the usability of whitelisting solutions largely depend on how *accurate* the underlying analysis is, as those solutions vary by the levels of granularity they achieve. At one extreme, there are course-grained whitelisting solutions that work like firewalls, controlling whether network traffic matches, for example, specific combinations of IP addresses and TCP ports. At the other extreme it is possible to devise fine-grained, *deep packet* whitelisting systems. These are capable of understanding the underlying protocol used by the application to be monitored and of blocking/detecting the invocation of a certain message type, or even the use of field values which do not respect pre-determined constraints.

The *accuracy* of the analysis largely determines the effectiveness of the whitelisting solution. For instance:

1. **Low Accuracy** Solutions whitelisting IP addresses/TCP ports combinations are capable of detecting/blocking traffic coming from e.g. an illegitimate source (access control), but cannot detect an advanced threat, or the misuse of an application. In fact, they are unable to detect the presence of malformed messages. They are also unable to detect the presence of messages triggering vulnerabilities or potentially harmful functionalities on the target device.
2. **Medium Accuracy** Some solutions understand the application protocol(s), and can whitelist protocol message types. These solutions can detect/block also attacks based on malformed communications or application misuses. For example, a medium accuracy whitelisting solution will be able to detect or block a message of type “stop the PLC”, but will be unable to detect/block a message whose field values trigger a buffer overflow vulnerability.
3. **High Accuracy** Solutions that, in addition, can whitelist message field values can detect/block also more complex attacks such as buffer overflows and data injection attacks.

In theory, solutions featuring *deep packet inspection* offer at least medium accuracy, though only DPBI (described below) offers high accuracy.

The main drawback of standard whitelisting solutions is constituted by their high configuration costs. The more accurate the desired level of analysis, the higher the set-up costs. **Whitelisting all message types and all message field values used in a production site is a daunting task that does not fit most budgets or time schedules.** This can be only partly mitigated by employing a course-grained configuration that fails to take advantage of the full power of deep protocol inspection, resulting in a marked reduction in coverage. Additional concerns and costs come from the fact that, each time a system like a PLC is reconfigured, updated or reprogrammed, the whitelisting solution that protects it must be likewise updated.

On the other hand, unlike anomaly detection solutions, whitelist solutions are amenable not only to detect, but also to *block* illegitimate traffic, thereby providing a higher level of protection. In practical applications, however, blocking comes at a high price:

- In order to avoid erroneously blocking of legitimate traffic, which could disrupt system availability, blocking engines are usually employed using a conservative configuration that does not take full advantage of the power of accurate whitelisting.
- A blocking whitelisting solution is inherently mission-critical: failure of a blocking solution such as a firewall would adversely affect the availability of the CI system it aims to protect, while rewarding the hacker with his desired result, disruption of service.
- In order to work properly, blocking systems need to be super-efficient in processing network data. Slowing down the network communication could adversely affect the availability of the CI, and is not acceptable. To achieve this efficiency, blocking systems have to compromise on the accuracy of the analysis. For this reason, the output of firewalls/IPS often consists of one-liners stating that they blocked a communication violating a certain rule. This gives too little ground for the forensics analysis needed to react to advanced threats.

In several documented cases a blocking whitelisting solution was employed as non-blocking to mitigate the possible availability risks, because the cost of disruption outweighs the advantage of automatic blocking of suspicious traffic.

3 A Novel Detection Method

As discussed above, traditional detection methods face significant challenges when striving to achieve their goal to defend networks. As a reminder:

- The weakness with blacklisting is that signatures do not provide coverage against zero-day and targeted attacks and attacks against Critical Infrastructure.
- The weakness in anomaly detection systems is their inaccuracy [1, 2], which makes them unsuitable for intrusion detection.
- The main weakness in whitelisting is that it is costly to deploy and maintain up-to-date. Basically, whitelisting solutions have to find a compromise between accuracy (detection rate) and configuration complexity (cost of ownership).

At SecurityMatters we have developed a detection method that makes up for the shortcomings of all of the traditional methods.

3.1 Deep Protocol Behavior Inspection (DBPI)

To combat the weakness of signatures, a solution needs to be non-signature based. Using a methodology we call *Deep Protocol Behavior Inspection*, it is possible to analyze and understand network traffic with a much higher degree of clarity compared to traditional detection methods. The advantage of DPBI is the ability to view and record the normal syntax and semantics of underlying network protocols.

In Appendix A and B we explain in more detail how this technology works; here we summarize its main features.

Hyper-accurate Deep Protocol Whitelisting DPBI enables the most accurate whitelisting that is possible to achieve given a protocol specification. We call it *Deep Protocol Behavior Inspection* rather than deep packet inspection also to stress the fact that the analysis does not stop at message types, as it is the case in the most accurate competing whitelisting solutions available today, but drills all the way down to, for instance, the value of each message

field. Each time a communication occurs, the DPBI-powered detection engine checks if it is protocol-compliant, if the message types are whitelisted, and *if the parameter values are appropriate*. The latter allows DPBI to detect a wide variety of attacks that are undetected by other solutions. Unlike anomaly detection systems, DPBI is also able to indicate with precision why a certain payload does not comply with the whitelisting rules, and provides operators and security analysts with accurate information upon which to make informed decisions.

Versatile DPBI is not pinned to a specific protocol: it is a general technology that can be instantiated to a binary protocol of choice: Modbus, MMS, OPC, SMB, ... All it takes is adding a specific module.

Learning, self-configuring, adaptive As discussed above, traditional whitelisting solutions need to find a compromise between analysis accuracy (detection rate) and configuration complexity (cost of ownership). DPBI enables the most accurate analysis possible thanks to a specific learning engine that generates the whitelisting rules automatically, thereby eliminating most of the configuration costs. At first, the DPBI-based learning engine listens to the network traffic and interprets it to infer the *Behavioral Blueprint* of the system. The Behavioral Blueprint is nothing else than a set of very accurate whitelisting rules, taking into account all network communication patterns, IPs and TCP ports, protocols, message types, message fields, and field values.

Whitebox The Behavioral Blueprint is not an obscure black box, but it consists of a set of easy-to-understand and easy-to-modify rules (see the examples in Appendix A). It can be grasped and edited by an operator or a security officer, for instance to add or remove specific checks. The DPBI-based detection engine is thus fully customizable, allowing for automatic as well as semi-automatic or full manual configuration, depending on the specific needs.

Current security technologies act like antibiotics, which can defend only against very specific and known diseases and bacteria. A single antibiotic will only be effective against a certain strain of bacteria. As bacteria evolve, to become resistant to the antibiotic, the antibiotic becomes useless. Antibodies, however, are effective on all strains of bacteria because they do not need to know the threat in advance; they detect antigens and neutralize illegitimate bodies as soon as they are encountered. DPBI is the cyber equivalent of antibodies.

DPBI provides access to a network's identity in such a way that provides a full understanding of network traffic, like an antibody understands DNA. Once the network's numerous characteristics are revealed to the system operator, they can be documented. After identifying the network's "DNA", it is then possible to flag as malicious any deviating data payload or activity. If a payload does not resemble the pre-recorded models, then actions can be automatically taken.

¹see e.g. [1, 2]

²Depends on the *depth* of the whitelisting analysis, and the type of attack.

³It is impossible to have a good set of attack signatures for ICS and PCN.

	Traditional Detection Methods			
	Blacklisting	Anomaly Detection	Whitelisting	DPBI
Chances of detecting an unknown attack	None	Low ¹	Medium ²	Very High
Suitable for Process Control networks	No ³	Not confirmed	Yes	Yes
Deep packet inspection	Yes	No, only done in lab systems	Sometimes	Yes
Deep protocol inspection	No	No	Theoretically possible	Yes
Set-up	Lightweight tuning	Automatic	Full manual configuration	Automatic set-up, manual tuning
Appropriate for blocking (IPS)	Yes, after careful tuning	No	Yes, after very careful tuning	Possible, but not implemented as design choice
Does not require updates to maintain maximum detection capabilities	No	Yes	Yes	Yes

Table 3.1: Technology comparison between DPBI and Traditional Detection Methods

4 SilentDefense ICS

SilentDefense ICS is a network monitoring and intrusion detection system based on DPBI, designed from the ground up to be applied to ICS.

Deep Analysis Upon installation, SilentDefense ICS employs the DPBI-based learning engine to analyze network traffic and to produce the Behavioral Blueprint of your network. The Behavioral Blueprint includes a wealth of information regarding the actual working of networked devices, including network communication patterns, commands and parameter used. It also clearly describes the access control patterns that are normal in your plant. With this information, SilentDefense ICS can raise an alert each time that user activity does not match the expected behavior. The Behavioral Blueprint also allows system operators to monitor variances in the network.

Broad Coverage Out of the box, SilentDefense ICS performs Deep Protocol Behavior Inspection of the following protocols: IEC 60870-5-101/104, DNP3, IEC 61850 (MMS and GOOSE), ICCP, OPC-DA, Modbus/TCP, CSLib (ABB's proprietary protocol), DMS (ABB's proprietary protocol), S7 (Siemens' proprietary protocol), EtherNet/IP, RPC/DCOM, and SMB/CIFS. More protocols will follow in the near future.

Access Monitoring: the smarter alternative to Access Control The benefits of access control systems are well understood: they help to prevent system misuse, data theft, and system downtime from an external or internal cyber attack. In practice, however, access control is often implemented either minimally or not all in ICS. This is due to the relatively high management costs and the risk that a misconfiguration of the access control system could jeopardize availability by blocking legitimate actions. SilentDefense ICS includes a self-configuring, non-blocking, customizable *access monitoring* system, flagging and reporting unusual network accesses as soon as they take place.

Do No Harm In the event of an emergency and you need to perform an undocumented action on your network, SilentDefense ICS works ideally because it doesn't block actions; it

reports them immediately to the security officer or system operator, allowing prompt reaction as well as forensics analysis. In an attack situation, SilentDefense ICS works well with third-party tools by helping them do their job better.

Low Management Costs SilentDefense ICS is self-configuring and self-learning. This allows network admins to cut costs by building a baseline model automatically and virtually zeroing set-up and configuration costs. This saves considerable time and money with respect to standard whitelisting solutions. Each time your system undergoes a reconfiguration it is not necessary to reconfigure the whitelist; SilentDefense ICS can be triggered to adjust itself automatically and adapt to the new situation. Since SilentDefense ICS uses non-signature based analysis, it does not require updates to guarantee full detection.

Future-proof Due to the modularity of the technology, SilentDefense ICS can be extended to cover virtually any new, customized protocol and proprietary solutions.

Combat Automated as well as Advanced Exploitation The flexibility provided by the Behavioral Blueprint allows SilentDefense ICS to detect automated exploitation tools and application-specific attacks – even when using proprietary protocols –, Layer-7 Denial of Service attacks, zero-day and advanced targeted attacks. While the former may be detected by traditional IDS/IPS as well, the three latter classes of attacks are not covered by those solutions at all.

4.1 Additional Benefits of SilentDefense ICS

Hyper-Accurate SilentDefense ICS does what IDS were meant to do, just a lot better. It detects known and unknown attacks at the time of entry, before they do any damage. In traditional systems, attackers can simply mutate their attack payload and fool IDS/IPS. Since SilentDefense ICS has no dependency on signatures, it will not be fooled by this tactic. With a detection rate of 99.9999% and a negligible false positive rate below .001%, SilentDefense provides positive assurance that the only activity on the network is your own.

Low-Impact and Super-Safe You could set SilentDefense ICS on fire and the availability of the rest of the plant would be unaffected. SilentDefense ICS is not installed on the host, is completely network-based, does not require any change to the underlying system, cannot add any additional risk to your devices, and is undetectable by hackers.

Quick, Inexpensive Deployment SilentDefense ICS is quick and inexpensive to set up, as it requires no change to the monitored system. SilentDefense ICS needs no manual configuration either, though some lightweight tuning and customization may be needed to perfectly fit your business and get the most effective detection rate. It can be up and running in a matter of 60 minutes and be fully operational within days or even hours in a medium sized plant.

Highly Customizable Using its whitebox architecture, SilentDefense ICS allows any level of tuning and an unprecedented degree of flexibility. The embedded Programmable Detection Interfaces (PDIs) let operators extend SilentDefense ICS' built-in capabilities and directly control the detection engine. For example, users can create custom detection policies tailored to their environment. Detection policies take advantage of the underlying DPBI technology to achieve extraordinary detail, even allow specific commands only at specified times, or from normally unauthorized network locations.

Compatible SilentDefense ICS is designed to integrate with other security solutions. It easily interfaces to most SIM/SIEM solutions, allowing seamless incorporation in present security solutions, even of competitors.

Broad Coverage, Extendible SilentDefense ICS currently covers the majority of the protocols commonly used in the CI industry. SecurityMatters is constantly adding new protocols to the learning and detection engines. This way, SilentDefense ICS can be adapted to protect specific custom systems like military missile control and shipboard management processes.

5 Case Studies

5.1 Threat #1: Chronic exposure to unknown software vulnerabilities

Industrial Control Systems are not targeted by mainstream attackers. Motivated, well-funded, and sometimes even state-sponsored organizations are behind successful intrusions. Attacks are carefully planned and executed employing specifically developed malware, which circumvents standard security solutions based on signatures. They exploit vulnerabilities previously unknown to system owners.

How SilentDefense ICS helps SilentDefense ICS employs a non-signature based approach, DPBI. The detection engine does not require prior knowledge of the nature nor the payload of an attack in order to detect it. By their nature, attacks aim to do something other than normal operations. Any communication pattern that deviates from the expected one will be immediately reported.

This immediate notification allows you to:

- be aware of a threat before it does any damage;
- avoid downtime by being able to take immediate action;
- know immediately, with detail, which systems are being attacked.

Real world Example SilentDefense ICS would have detected Stuxnet immediately after installation in two different ways. First, in the engineering network, the RPC interface used by Stuxnet to propagate would not match the Behavioral Blueprint. SilentDefense ICS would have reported the mismatch, pinpointed the computer that was infected and where the infection was coming from. Secondly, in the Process Control network, SilentDefense ICS would have detected the reprogramming of the PLC and notified the security officer.

5.2 Threat #2: Combat Internal Theft

Operators and engineers have thorough knowledge of all aspects of an industrial process. Because they may modify a system's workflow in the course of their normal work, they have access to sensitive information that could be passed on to third parties. Some industrial control systems have the ability to implement access control mechanisms, but organizations seldom use them to save on the costs of reconfiguration and to achieve a higher productivity/availability. Current PLCs, however, completely lack authentication mechanisms, making the need of access control even more compelling.

How SilentDefense helps The Behavioral Blueprint of SilentDefense ICS encompasses network communication patterns and user commands and gives you a fine-grained reference document for all network activity. Unusual commands, unusual parameter values in usual commands, and commands launched from unusual devices will not match the original Behavioral Blueprint and are immediately reported. SilentDefense ICS will learn automatically the access patterns and raise a flag when they are not complied with.

The monitoring capability of SilentDefense ICS will allow you to:

- reduce downtime loss by immediately detecting anomalous commands, and (even non-anomalous) commands issued from unexpected devices, regardless of whether the anomaly is due to a malicious user behavior or a human mistake;
- minimize costs of consequences and of restoring systems as offending commands are immediately detected and reported;
- reduce dramatically the costs of access monitoring, including real-time monitoring of remote and unmanned locations, by having a system that zeroes (re-)configuration costs;
- be able to inexpensively do extensive monitoring and compliance checks on your systems and your workers, also remotely.

Real world Example We deployed SilentDefense ICS in a real-life production site of a refinery. SilentDefense ICS generated a Behavioral Blueprint by observing network traffic across 50 hosts, including two OPC servers. The Behavioral Blueprint showed that normally operators query 5-10 variables per request. During the monitored period, however, an operator queried hundreds of variables per request. SilentDefense ICS detected and reported something out of the ordinary.

5.3 Threat #3: Downtime from malware

Despite strict network segmentation and corporate policies, Supervisory and Process Control networks are never truly isolated from the external world. External consultants, as well as support and maintenance personnel, are required from time to time to transfer data, for instance with USB sticks, thwarting network segmentation and policies. Malware can then enter and spread freely in sensitive environments.

How SilentDefense helps SilentDefense ICS' benefit is two-fold. First: at deployment time, it allows users to detect any inconsistency with the expected network activities. A typical example is a host communicating via uncommon ports or protocols, which can be due to a misconfiguration or to the presence of malware on the host. Second: once fully operational, SilentDefense ICS detects any deviations from expected network communication patterns. Malware usually spreads leveraging network shares or vulnerable network services. When network communication patterns do not match the original Behavioral Blueprint, such

occurrences are reported to security analysts for manual or automatic action. SilentDefense ICS allows the operator to:

- take early action and contain the attack surface;
- minimize costs associated to production loss;
- cut downtime;
- use the Behavioral Blueprint to recall normal functions while restoring the system.

Real world Examples An energy company was hit by malware (ICS-CERT newsletter Jan 2013) and was forced to take its system down for 3 weeks in order to restore normal operations. If SilentDefense ICS were installed, it would have detected the anomalous network communications in seconds.

The Shamoon malware hit 30,000 workstations at Saudi Aramco. The disruption required a costly restore process. In presence of this kind of malware, security officers need to discover the compromised hosts first (a long process, in case of a widespread virus) and then restore them. SilentDefense ICS would have detected the anomalous access to remote shares used by Shamoon, allowing security officers to contain the viral spread. SilentDefense ICS would have also indicated with precision which hosts may have been compromised, allowing a prompt and efficient recovery process.

5.4 Threat #4: System configuration in production shifts from expected blueprint

Any organization dealing with sensitive industrial process tests system configurations prior to deployment and stores a copy (a blueprint) of running configurations in case of disruption, so that it can be restored as quickly as possible. Running configurations tend to change over time, for instance, due to on-site maintenance and adjustments. Blueprints should be updated as well. When blueprints are not kept up-to-date, the restore procedure will not be effective and cause additional delays and prolonged downtime.

How SilentDefense helps SilentDefense ICS' Behavioral Blueprint can be employed to automatically check for consistency between operational and test environments, thereby lowering the risk of additional downtime due to a flawed restore blueprint.

SilentDefense ICS allows network admins to:

- reduce costs associated with system downtime;
- reduce costs associated with system restore;
- guarantee compliance between installations;
- reduce operational costs since it enables employees to understand and report on the actual traffic present in networks.

Real world Example We deployed SilentDefense ICS in a production site of a gas storage facility and generated the Behavioral Blueprint of a Process Control environment, where MMS and SMB protocols were used. The Behavioral Blueprint comprised communication patterns (IP addresses and TCP ports), as well as protocols used among hosts. To test SilentDefense ICS against network attacks without affecting the production environment, we moved the Behavioral Blueprint to a mockup test environment that was expected to be a twin copy of the actual production site. Once the Behavioral Blueprint was instantiated in the test environment, several discrepancies were automatically detected and highlighted for review.

Several devices had different IP addresses and were using different protocols than expected, as they had not been observed before.

6 Closing

After almost twenty man-years of research on existing security solutions, their shortcomings and the new threats landscape, SilentDefense ICS answers the combined problems of zero-day threats, highly advanced attack profiles, interconnected and standardized systems, and usability issues.

SilentDefense ICS' advanced technology directly addresses the urgent need for protecting CI organizations from a wide array of highly evolved attacks, no matter where or how they originate.

To enable zero-day defense and overall network integrity, the Behavioral Blueprint of SilentDefense ICS shows a user information not available in any other way, including communications between devices, unusually high level of queries/activities, anomalies in the network, differences between production and testing environments, and more.

SilentDefense ICS is scalable and highly-configurable, quick to set up and has operational benefits that support the highest security goals as well as operational and cost-saving measures.

6.1 Acknowledgments

We thank Daniel Trivellato, David Robinson, Jeff Taylor, Bob Youngblood, and Marcel Jutte for their constructive comments and improvements.

6.2 About SecurityMatters

SecurityMatters is a spin-off from the Dutch Technical Universities. Its research team works with many different projects throughout the EU and USA and delivers game-changing network monitoring and intrusion detection technology to make their customers more secure and in control.

Some History When Damiano Bolzoni and Emmanuele Zambon left Italy and their jobs to join the research group of Sandro Etalle in the Netherlands, they had the declared intention to “create the most innovative and effective intrusion detection system of the world and build an enterprise to bring it to the market”. After years of research in which several different technologies have been devised and tested, DPBI saw the light. It was a truly new, truly effective way of realizing network monitoring and intrusion detection. SecurityMatters was founded in 2008 and incorporated in December 2009. Under Emmanuele’s lead, SecurityMatters development team built SilentDefense around DPBI. Version 1.0 was ready and first placed on the market in September 2011. In 2012, SecurityMatters won the SecurityMatters wins the COMMIT Wetenschapsvalorisatieprijs.

6.3 About the Authors

Sandro Etalle (1965) Sandro Etalle was 19 when he founded his first high-tech company. Meanwhile, he finished his studies at the conservatory (flute) and in mathematics (cum laude). In 1992, he left the business world to pursue an academic career. In 1995, he received a PhD from the University of Amsterdam, and is now also full professor and head of the computer security group at the Technical University of Eindhoven. He is one of the co-founders of SecurityMatters. As scientist, Etalle wrote more than 100 articles cited more than 1500 times.

Cliff H. Gregory (1948) Prior to his work in the private sector, he served 28 years in the U.S. Navy, including assignments developing cyber security programs. His experience in Agile software product and process development, as well as business process thought leadership give him a unique vision of how to improve the safety and security of IT organizations. A published author and accomplished public speaker and trainer, Cliff holds a PhD in Math from the Dublin Institute of Technology (DIT) and a number of IT and security certifications. During his more than 35-year career in both government and private industry, Cliff has focused on Information Assurance and Systems Integrity.

Damiano Bolzoni (1981) Damiano Bolzoni graduated in 2005 in Computer Science from the Ca’ Foscari University of Venice with a thesis on anomaly-based network intrusion detection system, together with his fellow classmate Emmanuele Zambon. During his master studies he worked for the Information Risk Management division of KPMG Italy. He then moved to the Netherlands to pursue a PhD at the University of Twente. He was member of the Italian Under 20 Athletics team and took part in several international competitions.

Emmanuele Zambon (1980) Emmanuele Zambon has been involved in computer security since 2003. In 2005, he also had graduated in Computer Science from the Ca’ Foscari University of Venice with a thesis on Intrusion Detection Systems. During and after his studies he has been employed in the Information Risk Management group of KPMG and in Telecom Italia as a consultant, doing ethical hacking, network vulnerability/assessment, and software development/performance tuning. In 2006, Zambon followed Bolzoni to the Netherlands to work together on the development of a new technology (the core of which now forms SilentDefense). Zambon received his PhD in IT Risk Management in 2011 from the University of Twente. He is now working part-time as a post-doc at the University of Twente, doing research on intrusion detection for industrial process automation networks, and working part-time for SecurityMatters.

Bibliography

- [1] D. Hadziosmanović, L. Simionato, D. Bolzoni, E. Zambon, and S. Etalle. In *Proceedings of the 15th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2012), Amsterdam, The Netherlands*, volume 7462 of *Lecture Notes in Computer Science*, pages 354–373, 2012.
- [2] Y. Song, M. E. Locasto, A. Stavrou, A. D. Keromytis, and S. J. Stolfo. On the infeasibility of modeling polymorphic shellcode. In *Proceedings of the 14th ACM conference on Computer and communications security, CCS '07*, pages 541–551. ACM, 2007.

Appendix A Understanding DPBI

DPBI combines the advantages of the most accurate whitelisting technology with the ease-of-use of a fully customizable, self-configuring whitebox system.

DPBI operations are divided in three phases: the learning phase, the tuning and customization phase, and the detection phase.

A.1 The Learning Phase

During this phase, the DPBI-based learning engine inspects the network traffic and infers its Behavioral Blueprint, which contains information regarding the applications running on the monitored systems at different levels of granularity. For instance, at IP level, it records which IP addresses and IP ports are in use, and which communication patterns are common. Furthermore, thanks to the DPBI technology, the learning engine understands the semantics of the protocol in use. As a result, the Behavioral Blueprint includes an inventory of message types, message fields, and field values in use by the applications in *your* system. In other words, DPBI understands in detail how the monitored system actually works, including details like the common settings of your PLCs.

How long the learning phase takes depends on how complex the underlying network traffic is. For a simple, Modbus-based network the training phase is shorter than for a more complex network with more protocols such as MMS, IEC 104, OPC-DA. In one practical case, we had excellent results with a learning phase of just 5 minutes, though a couple of days is more common.

If for some reasons the Behavioral Blueprint is not accurate, for instance because a PLC has been reprogrammed after the learning phase, then the Behavioral Blueprint can be easily tuned in several ways (see tuning and customization below).

A.2 The Tuning and Customization Phase

Throughout the whole process, at any time, the system can be tuned and customized.

The Behavioral Blueprint is not black magic, on the contrary: it is presented to the user as a set of easy-to-understand rules that can also be employed to understand what actually happens in your network, providing a phenomenally accurate, easy-to-use report that can be used, e.g., for compliance checks (see case studies).

```

1 hosts $SCADA_SERVERS { 172.16.4.1, 172.16.4.11, 172.16.4.21, 172.16.4.71, 172.16.80.21 }
2 hosts $PLCS { 172.16.80.151 }
3
4 ports $OPC { 1025-65535 }
5 ports $DCOM { 1825-85535, 135 }
6 ports $SMB { 139, 445 }
7
8 class NotAKnownOne 0 { { * } }
9 class MMR 0 { { mt { 4, 5, 31, 805, 1105 } } }
10 class OPC-DA 0 { { mt { 6, 33554430, 33554480, 33554482, 33554532, 33554884, 33554932, 33555433, 33555438, 33555439 } } }
11 class LDAP 0 { { * } }
12 class DCOM 0 { { mt { 0, 2, 11, 14, 15, 16 } }, { if { { 0000143-0000-0000-c000-000000000046, 3 }, { 0000143-0000 } } } }
13 class SMI 0 { { smbl-mt { 4, 37, 43, 45, 46, 47, 50, 113, 114, 115, 116, 117, 160, 162 } }, { smbl-mt { } }, { mt } }
14 class NetBIOS 0 { { * } }
15 class HTTP 0 { { * } }
16
17 alert [ * ] { [ * ] -> [ * ] { [ * ] using { TCP } with { * } }
18 allow [ * ] { [ * ] -> [ $SCADA_SERVERS ] = { 3389 } using { TCP } with ( { class NotAKnownOne 0 } )
19 allow [ $SCADA_SERVERS ] = { * } -> [ $SCADA_SERVERS ] { [ $DCOM ] using { TCP } with ( { class DCOM 0 } ) }
20 allow [ $SCADA_SERVERS ] = { * } -> [ $SCADA_SERVERS ] { [ $OPC ] using { TCP } with ( { class OPC-DA 0 } ) }
21 allow [ $SCADA_SERVERS ] = { * } -> [ $SCADA_SERVERS ] { [ $SMB ] using { TCP } with ( { class SMB 0 }, { class NetBIOS 0 } ) }
22 allow [ $SCADA_SERVERS ] = { * } -> [ 172.16.4.1 ] { 88 } using { TCP } with ( { class NotAKnownOne 0 } ) }
23 allow [ $SCADA_SERVERS ] = { * } -> [ 172.16.4.1 ] { 389 } using { TCP } with ( { class LDAP 0 } ) }
24 allow [ $SCADA_SERVERS ] = { * } -> [ 172.16.4.11 ] { 80 } using { TCP } with ( { class HTTP 0 } ) }
25 allow [ $SCADA_SERVERS ] = { * } -> [ $PLCS ] = { 202 } using { TCP } with ( { class MMR 0 } ) }

```

Figure A.1: Example of customizable rules inferred by SilentDefense ICS (non protocol-specific) for network connections.

The Behavioral Blueprint can be fully edited by the user, at any time. Like all whitelisting systems, the Behavioral Blueprint may at a certain point in time require updating, for instance when PLCs have been reprogrammed.



The screenshot shows a tree view of protocol message rules on the left and an 'Edit Leaf node' dialog box on the right. The tree view shows a hierarchy of nodes with their respective counts: application_layer (7183), request (7183), header (7183), data (7183), confirm (624), select (3265), operate (3265), read (29), object_header (29), object_type_field (29), qualifier_field (29), range_field (29), and no_range_field (29). The 'Edit Leaf node' dialog box shows the following fields: Node name: len, Comment: (empty), Model type: Numeric unsigned model, Active testing type: Range, Sample count: 7183, Policy: Grow Default, Allowed range: Min: 8 Max: 16, Range: Min: 8, Max: 16, and buttons for Cancel and Finish.

Figure A.2: Example of the customizable rules inferred by SilentDefense ICS (protocol-specific) for protocol messages.

In addition, the detection engine can be instructed through an easy-to-use script language to raise alerts when very specific events take place. Examples of such events include: when a certain message type is used, when a certain field value matches user-defined constraints, when a certain function is not used for too long, and when a connection to a PLC is made from a specific location.

SilentDefense ICS copes easily with small and big changes in the monitored system: for instance, when a new message type is detected, an alert may be raised. At this point the security officer may decide to include the new message type into the Behavioral Blueprint. All it takes is pushing a button. When more radical changes take place in your system, you may decide to extend or restart the learning phase. This is easy to do, and requires no effort

from the user side.

Finally, the level of accuracy of the detection engine is also fully customizable. For instance, the engine can be instructed to stop the analysis at a certain level in the protocol tree, or to disregard a certain part of the protocol altogether.

A.3 The Detection Phase

After the tuning and customization phase, the detection phase can start. In this phase, anything that deviates from the inferred Behavioral Blueprint is reported to the security officer.

Thanks to the fact that the analysis is *deeper* than in any other whitelisting solution, SilentDefense ICS detects attacks such as data injection attacks that no other solution would detect. And since the Behavioral Blueprint includes communication patterns, system misuses like the issuing of a command from an illegitimate location are also reported, making SilentDefense ICS an effective access monitoring platform.

The number of protocols covered by SilentDefense ICS is higher than any competitor and includes both ICS and Windows-based protocols. As a result, SilentDefense ICS can be used throughout the whole network, including: back-office, Supervisory, and Process Control network, providing a uniform solution for today's security challenges.

The detection engine easily interfaces with widespread SIM/SIEM solutions. This way, SilentDefense ICS can be integrated with present security solutions, even of competitors.

Appendix B Frequently Asked Questions about DPBI

Q: What is the difference between DPBI and “deep packet inspection whitelisting”?

A: “Deep packet inspection whitelisting” indicates a technology that understands the underlying application protocol syntax and semantics and that can do whitelisting of some protocol features. The accuracy of the inspection, however, depends on the depth of the analysis. In practice, deep packet inspection typically stops at the message type (e.g. Modbus/TCP “read” and “write” functions), while DPBI includes also parameter values, allowing to detect a whole extra class of attacks, and to carry out the most accurate custom checks. DPBI can be regarded as a combination of the most accurate deep packet inspection possible, with a learning engine and a fully customizable detection engine.

Q: To use SilentDefense do I have to go through a learning phase for each network in my system, or can I upload my own, custom-made Behavioral Blueprint?

A: You do not have to go through a learning phase, and you may decide to configure the system manually – as our competitors do. This is possible thanks to the simple syntax and the 100% customizable character of the Behavioral Blueprint. Practical experience, however, shows that it is far easier and less time-consuming to use the learning engine – possibly tuning the learned Behavioral Blueprint afterwards – than doing a full manual configuration.

Q: What happens if an attack takes place during the learning phase? Will the resulting Behavioral Blueprint be unable to detect such attack later on?

A: In theory, if an attack takes place during the learning phase the attack data could be included in the Behavioral Blueprint and the successive instances of the attack could remain undetected. However, practice shows that this is really not an issue, even in systems connected to the Internet, in which attacks happen all the time. In fact, SilentDefense ICS

incorporates mechanisms that allow to exclude spurious network events from the inferred Behavioral Blueprint.

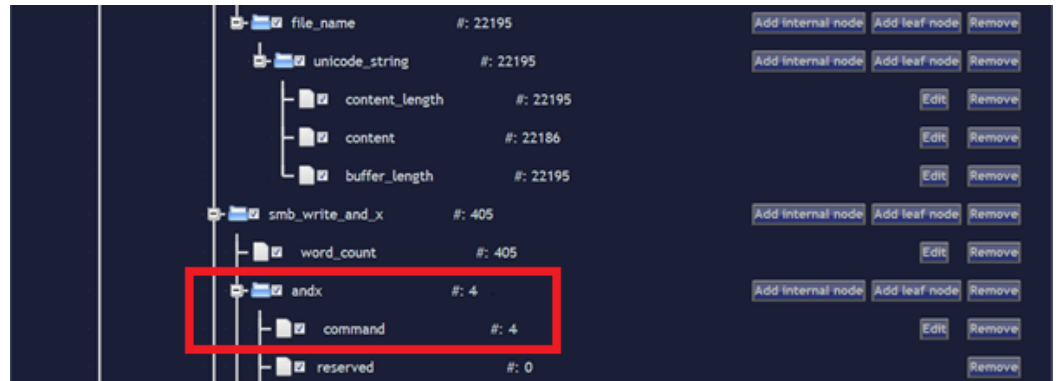



Figure B.1: Example of Behavioral Blueprint for the SMB protocol. Counters indicate the number of times a field has been observed during the learning phase.

Q: How do I understand when the learning phase is complete?

A: Understanding when the learning phase is complete is fairly straightforward: it is when all normal network operations have been performed and you don't see new information being added to the Behavioral Blueprint.

Q: Do I need full knowledge of network protocols to understand the cause of an alert?

A: The detection engine provides extremely detailed information about unusual network events, down to the content of anomalous message fields and field values. This does not mean, however, that in order to understand an alert you need full knowledge of the underlying network protocol. In fact, by attaching expressive labels to alert information, SilentDefense ICS enables you to easily understand the cause of an alert. For example, it will tell you whether the alert was due to a number of message fields which are not part of the Behavioral Blueprint, and provide you a list of those fields, or whether some fields have unusual values, and allow you to compare those values to the normal ones. In addition, thanks to its classification engine, SilentDefense ICS helps you prioritizing alerts and suggest whether an unusual event resembles a known attack vector. Intuitively, in order to determine whether a certain event represents a new, unknown attack, a more in-depth and experienced analysis of the alert is required. But enabling you to detect and analyze new and unknown attacks with such a level of detail is a feature that no other present security solution can offer.



Alert Detail Type:	DPBI
Direction:	Upstream
DPBI Detail Type:	Unusual Field Value
L7 Protocol:	SMB
Path To The Field:	/ body / session_message / smb_pdu / smb1_pdu / request / smb_header / flags2
Field Name:	flags2
Unusual Sample:	10241
Model Data:	{49219, 51201, 51207, 51283, 55303, 59399} - samples: 58,461
Trim the model using this alert detail	<input type="button" value="Trim"/>

Figure B.2: Example of alert detail. The cause of the alert is an unusual value for field “flags2” in the body of a SMB message. The value (10,241) does not belong to the set of learned values for the field.