A Trend Micro Research Paper

The SCADA That Didn't Cry Wolf

Who's Really Attacking Your ICS Equipment? (Part 2)

Kyle Wilhoit
(Trend Micro Forward-Looking Threat Research Team)

Contents

TREND MICRO LEGAL DISCLAIMER

The information provided herein is for general information and educational purposes only. It is not intended and should not be construed to constitute legal advice. The information contained herein may not be applicable to all situations and may not reflect the most current situation. Nothing contained herein should be relied on or acted upon without the benefit of legal advice based on the particular facts and circumstances presented and nothing herein should be construed otherwise. Trend Micro reserves the right to modify the contents of this document at any time without prior notice.

Translations of any material into other languages are intended solely as a convenience. Translation accuracy is not guaranteed nor implied. If any questions arise related to the accuracy of a translation, please refer to the original language official version of the document. Any discrepancies or differences created in the translation are not binding and have no legal effect for compliance or enforcement purposes.

Although Trend Micro uses reasonable efforts to include accurate and up-to-date information herein, Trend Micro makes no warranties or representations of any kind as to its accuracy, currency, or completeness. You agree that access to and use of and reliance on this document and the content thereof is at your own risk. Trend Micro disclaims all warranties of any kind, express or implied. Neither Trend Micro nor any party involved in creating, producing, or delivering this document shall be liable for any consequence, loss, or damage, including direct, indirect, special, consequential, loss of business profits, or special damages, whatsoever arising out of access to, use of, or inability to use, or in connection with the use of this document, or any errors or omissions in the content thereof. Use of this information constitutes acceptance for use in an "as is" condition.

Introduction

"Who's Really Attacking Your ICS Equipment?" presented a thorough outline of a honeynet specifically developed to catch attacks against industrial control systems (ICS).[1] The devices featured in the paper were external facing and riddled with vulnerabilities commonly found plaguing ICS equipment worldwide.

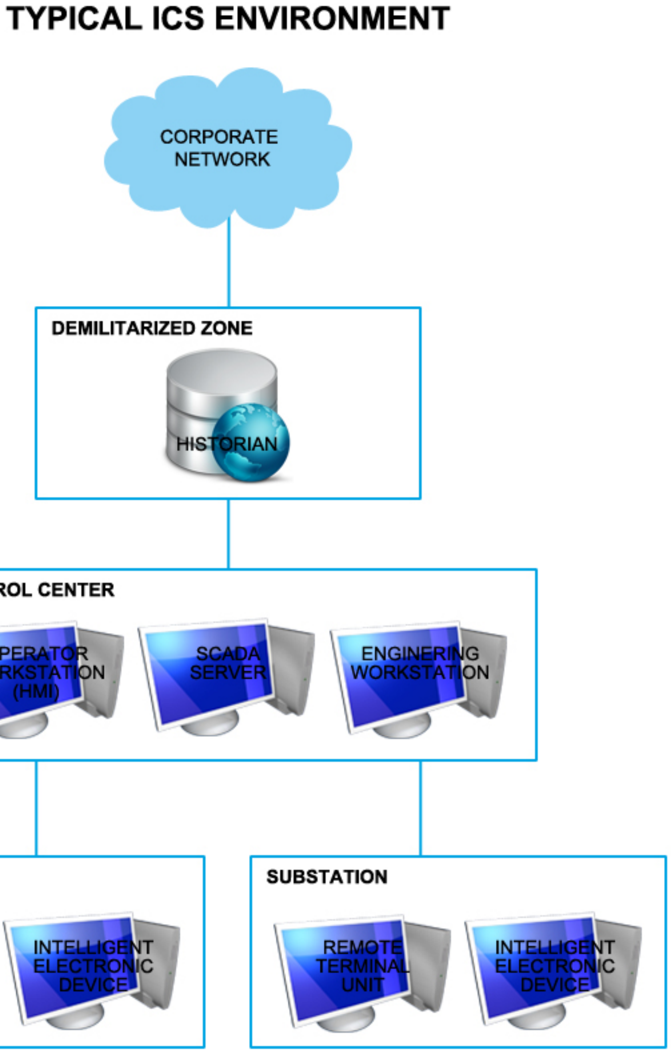
Supervisory control and data acquisition (SCADA) networks are systems and/or networks that communicate with ICS to provide data to operators for supervisory purposes as well as control capabilities for process management. As automation continues to evolve and becomes more important worldwide, the use of ICS/SCADA systems is going to become even more prevalent.

In this paper, we looked at who are continuing to attack external-facing ICS devices and why. It also features a more robust honeynet architecture we developed and deployed worldwide over a period of months. This paper intends to fully showcase not only attack statistics but also show the robust attribution framework we utilized. Finally, it includes more in-depth analysis of the threat actors and their possible motivations behind attacks.

[1] Kyle Wilhoit. (2013). "Who's Really Attacking Your ICS Equipment?" Last accessed June 27, 2013, http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-whos-really-attacking-your-ics-equipment.pdf.

Traditional Honeypot Deployments

When referencing our honeypot deployments, it is very important to understand how a traditional ICS environment looks.



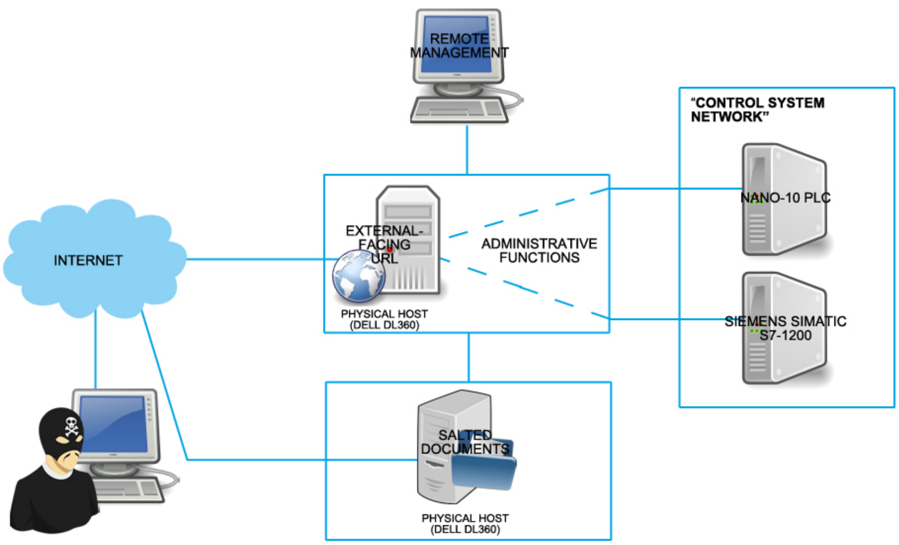
As shown above, no security devices or protocols appear to be in place. While some ICS environments do contain these measures, most do not have such preventative controls.

Original Honeypot Deployment

"Who's Really Attacking Your ICS Equipment?" featured an external-facing honeypot deployment stationed in the United States, which was specifically designed to attract targeted attacks. The honeypot architectures were of two distinct types—high and low interaction.

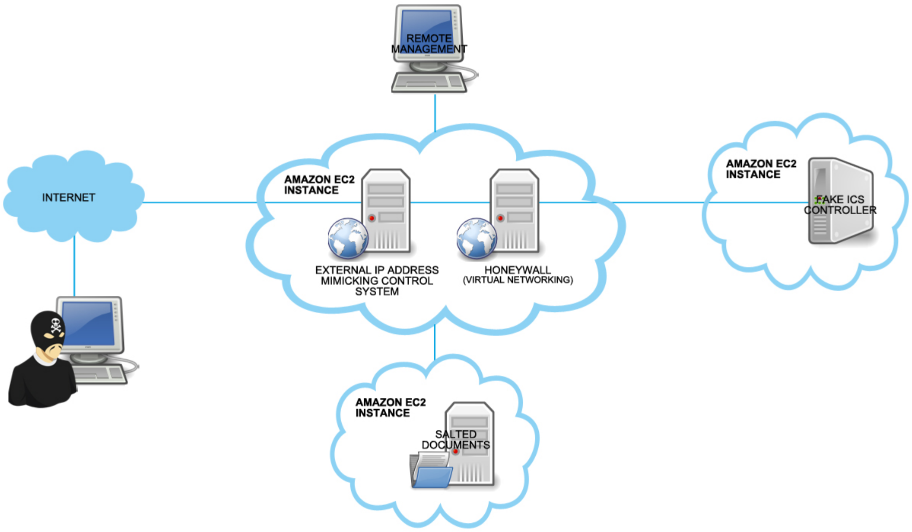
The first honeypot was a high-interaction one, which imitates the activities of a physical ICS device.

ORIGINAL HIGH-INTERACTION HONEYPOT



Apart from high-interaction honeypots, we also used a low-interaction architecture. Low-interaction honeypots can be characterized as "traps used to simulate the services provided by a production system." These honeypots use very little resources and allow multiple instances to be virtually spun up if desired.

ORIGINAL LOW-INTERACTION HONEYPOT



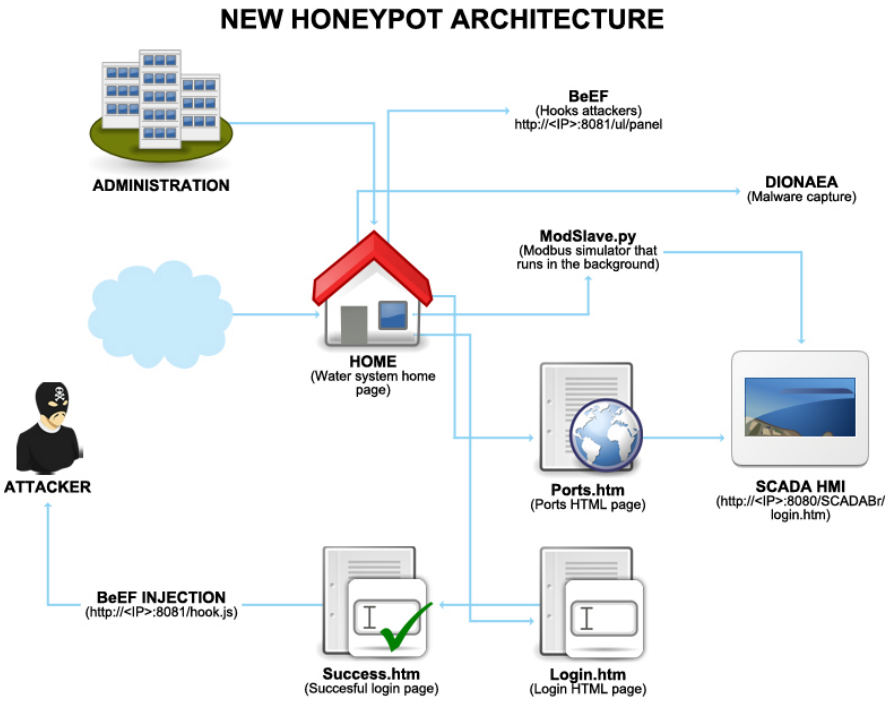
New Honeypot Architecture

While the original honeypot deployment was successful and garnered accurate results, we wanted a bigger data sample to better represent the global perspective. So, a more robust virtualized environment that could be deployed in a matter of minutes anywhere in the world was created. We continued to stay with the ruse of being municipal water control systems worldwide. Traditionally, municipal water districts, even worldwide, have very little control over the systems that run municipal water supplies. These systems are traditionally rarely secure, which is why we continued to choose them for our purpose.

The first challenge in rearchitecting the honeypot solution was to create a believable, fully mimicked version of a virtualized ICS environment. To accomplish this, newly created tools and already-existing toolsets were utilized across multiple virtualized environments.

The second challenge was to create a full-featured service emulation module, which was also accomplished via the utilization of newly created and already-existing tools and scripts.

Other challenges had to do with attack attribution. Attributing attacks using IP addresses is very inconsistent and provides little value to an organization that wants to know who attacked it. As such, each of the honeypots utilized in the architecture used a module called "The Browser Exploitation Framework (BeEF)," which helped attribute attacks to a particular attacker or group of attackers.[2]



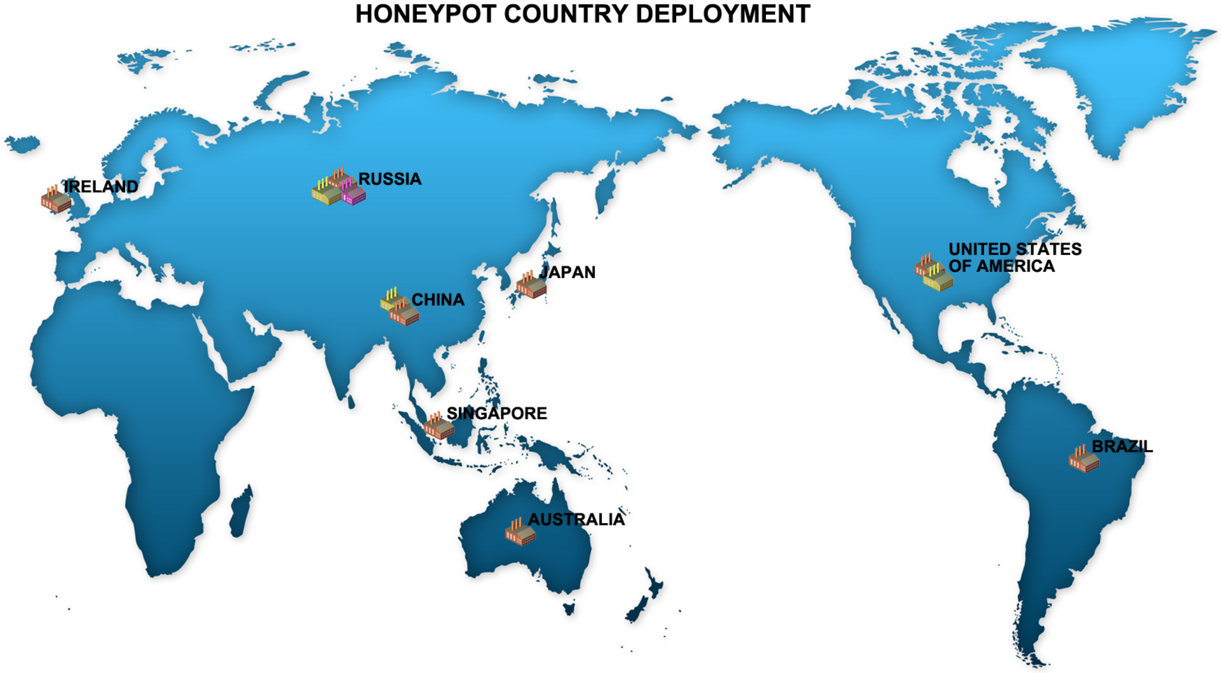
When looking at the new honeypot architecture, it is easiest to think of each section as a "module" that independently operates from the others. Many of these modules operate on a single virtual machine, except the human-machine interface (HMI), which operates on a logically separated virtual instance. In addition, the programmable logic controller (PLC) device with which the HMI interacts is also logically separated from the other devices.

[2] *BeEF*. http://beefproject.com/.

New Honeypot Deployment

Because we aim to continue gathering realistic targeted attack scenarios worldwide, we first had to virtualize and make our honeypot architecture more robust. After doing this, we focused on multiplying and expanding our number of honeypots to turn our architecture into a honeynet. Note that a honeynet is a network of honeypots that is traditionally distributed geographically. In this case, however, all of our honeypots worked separately and did not communicate with one another in any fashion. Segregation ensured that no cross-communication contamination would occur in case an attacker compromises a single honeypot on our honeynet.

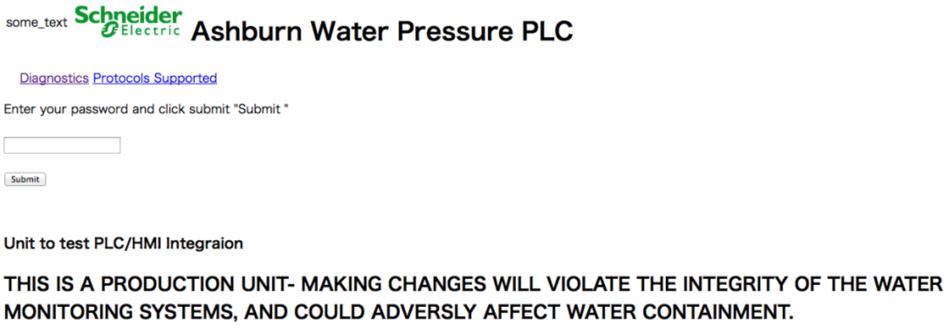
HONEYPOT COUNTRY DEPLOYMENT



The diversity of the countries we chose to deploy honeypots in helped generate a wide coverage of attacks. The country breakdown should help you visualize where the honeypots were deployed.

Honeypot Country Deployment	
Honeypot Location	**Number Deployed**
China	2
Japan	1
Russia	3
Australia	1
USA	2
Ireland	1
Brazil	1
Singapore	1
Total	**12**

In addition to deploying honeypots worldwide, we also made sure we localized all of the text in the honeypot deployments, depending on where they were located. This proved to be an arduous task that required the help of research colleagues familiar with the languages and customs local to the honeypot location.



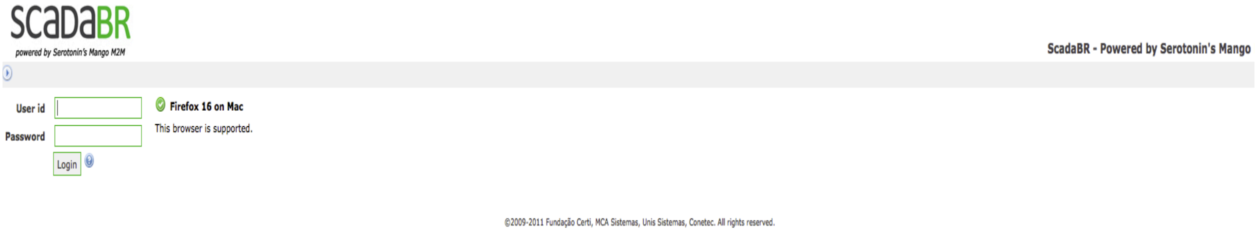
some_text Schneider Electric Ashburn Water Pressure PLC

Enter your password and click submit "Submit "

Unit to test PLC/HMI Integraion

THIS IS A PRODUCTION UNIT- MAKING CHANGES WILL VIOLATE THE INTEGRITY OF THE WATER MONITORING SYSTEMS, AND COULD ADVERSLY AFFECT WATER CONTAINMENT.

Sample main web page of a honeypot instance



ScadaBR - Powered by Serotonin's Mango

User id

Password

Firefox 16 on Mac

This browser is supported.

©2009-2011 Fundação Certi, MCA Sistemas, Unis Sistemas, Conetec. All rights reserved.

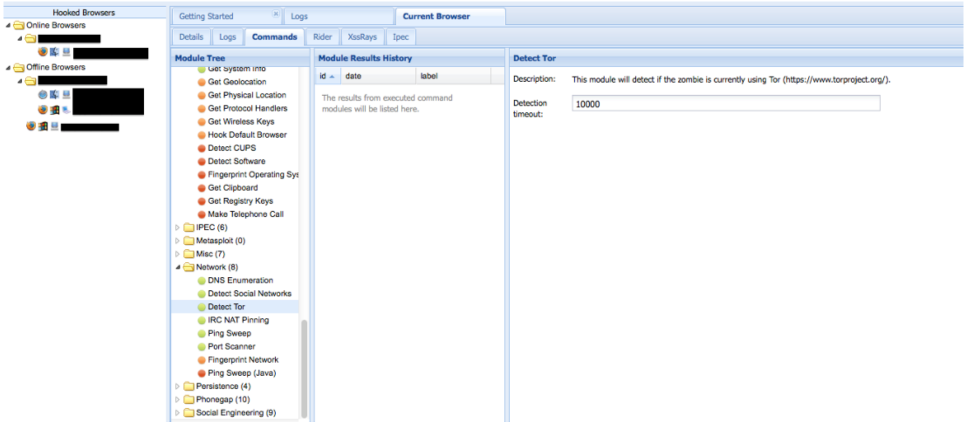
Sample HMI page

Attribution Framework

Determining an attacker's location based on the IP source address of incoming connections is inconclusive. Attackers often use anonymizers like Tor to change their source IP addresses.[3]

To help combat attribution-related issues, an excellent framework—BeEF—was used. While the use of this framework could be nefarious by nature, when used properly, it allows security researchers and analysts to more effectively attribute attacks in greater detail.

BeEF, as a framework, can actively run scripts on a victim's browser every time the user accesses a certain web page. A BeEF injectable script was embedded into a web page that could only be accessed using secure credentials stored in the honeypot environment. The page was in the honeypot architecture behind a secure area. As such, a potential victim must access the page inside the secure area in the honeypot for his/her browser to be affected. So, if an attacker compromises website authentication, BeEF would run the script to help determine his/her geographical location and obtain other statistical data.



The results from executed command modules will be listed here.

Description: This module will detect if the zombie is currently using Tor (https://www.torproject.org/).

Detection timeout: 10000

Sample BeEF administration portal

Within BeEF, the get physical location module will retrieve geographical location information based on neighboring wireless access points using commands encapsulated within a signed Java applet. The get system info module will, meanwhile, pull system information using an unsigned Java applet. The data obtained includes operating system (OS) details, number of processors, NIC names and IP addresses, along with other details. Finally, the detect Tor module will detect if the machine used runs Tor.

[3] The Tor Project, Inc. *Tor.* Last accessed July 3, 2013, https://www.torproject.org/.

Apart from BeEF, several other attribution methods and internal tools were used. While we cannot specifically share what these methods are, we are confident that the correlation between BeEF and our internal tools can help determine an attacker's physical location very well.

Attacks

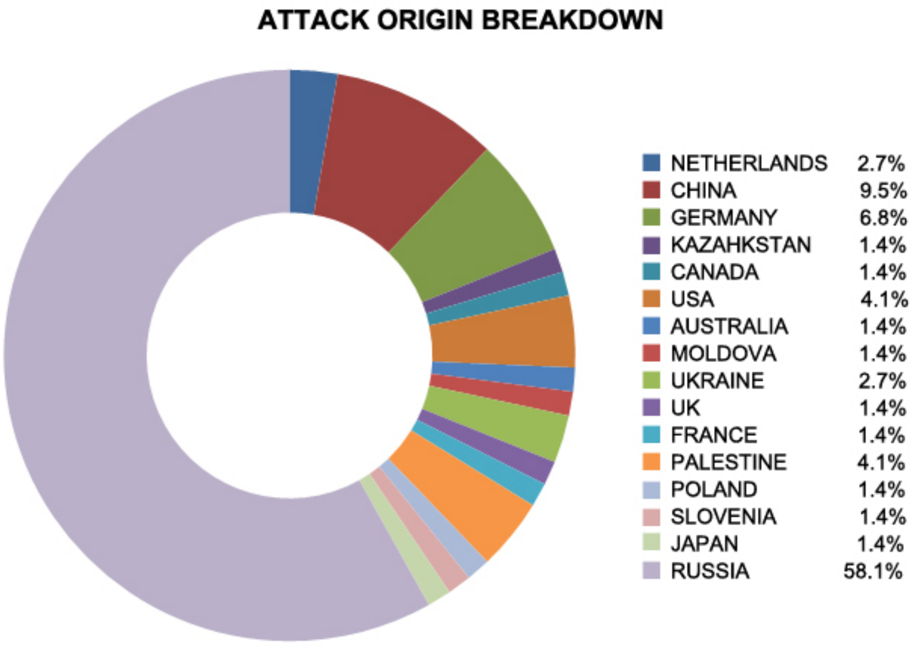
ICS attackers can often be likened to traditional targeted attackers. In the course of conducting research, we have seen ICS attackers take the same steps as targeted attackers do prior to staging attacks. Many perform reconnaissance not just on their target IP addresses but also on the netblock where the devices are hosted, which is traditionally seen in a /24 network. This stage typically involves port scanning of surrounding subnets. The attackers also perform fingerprinting on devices to ascertain their OSs, if possible, along with other identifiable information. They traditionally identify vulnerabilities at this stage as well. Once access to devices is gained, persistence and lateral movement were also observed in roughly 70% of the attacks we witnessed. Data exfiltration is also commonly seen. In one particular instance, we were able to actively witness the exfiltration of perceived virtual private network (VPN) configuration files leaving the compromised server.

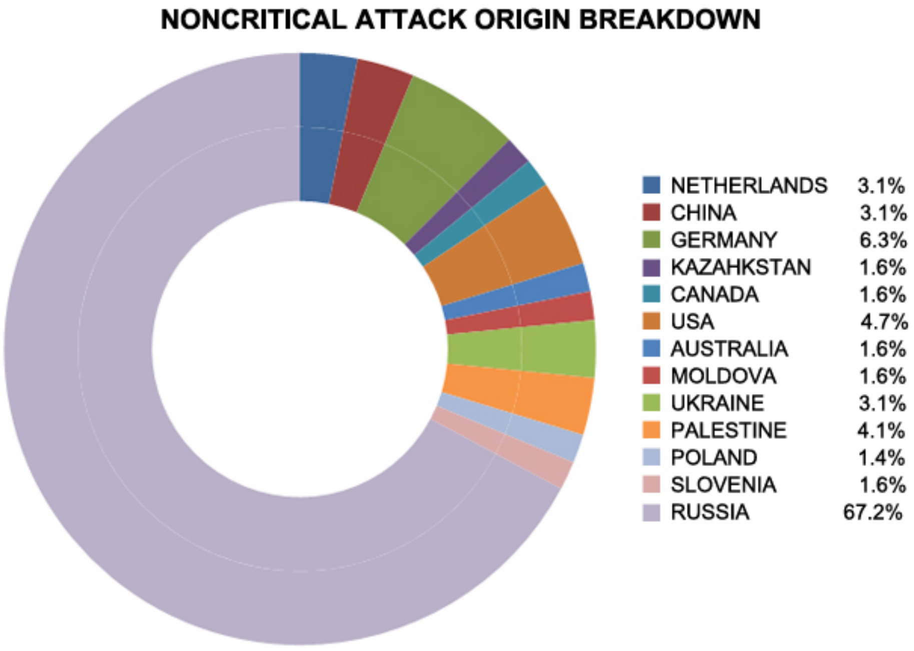
Over a period of three months, several attacks took place. Some were even able to compromise the entire operation of an ICS device. While many would consider an attack to be any type of drive-by or automated attack (e.g., "mass" SQL injection), we did not consider this type in this research paper. We only accounted for attacks that were considered targeted in nature (i.e., showed that a reasonable amount of reconnaissance was done prior to engaging in fingerprinting or the actual attack).

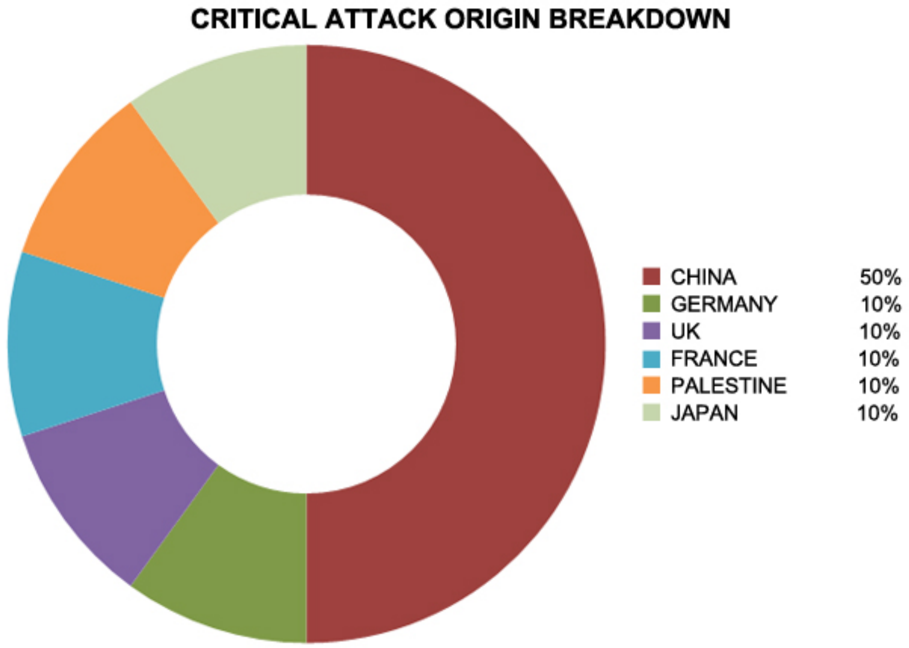
From March to June 2013, we observed attacks originating from 16 countries, accounting for a total of 74 attacks on seven honeypots within our honeynet. Out of these 74 attacks, 10 were considered "critical." When we refer to attacks as critical, we are referring to those without established motivations but can cause the catastrophic failure of an ICS device's operation. Likewise, attacks considered noncritical cannot cause a catastrophic failure but should they continue can. These types of attacks can take the form of a distributed denial-of-service (DDoS) attack, for instance.

Country of Orgin	Non-Critical Attacks	Critical	Total
Netherlands	2	0	2
China	2	5	7
Germany	4	1	5

Kazahkstan	1	0	1
Canada	1	0	1
USA	3	0	3
Australia	1	0	1
Moldova	1	0	1
Ukraine	1	0	1
UK	0	1	1
France	0	1	1
Palestine	2	1	3
Poland	1	0	1
Slovenia	1	0	1
Japan	1	1	2
Russia	43	0	43
Totals	**64**	**10**	**74**







More details on how the attacks are broken down by type are shown in the following table.

Attack Origin and Type Breakdown			
Country	Type		Total
	Critical	Noncritical	
Netherlands	2	0	2
China	2	5	7
Germany	4	1	5
Kazahkstan	1	0	1

Attack Origin and Type Breakdown

Country	Type		Total
	Critical	Noncritical	
Canada	1	0	1
USA	3	0	3
Australia	1	0	1
Moldova	1	0	1
Ukraine	2	0	2
UK	0	1	1
France	0	1	1
Palestine	2	1	3
Poland	1	0	1
Slovenia	1	0	1
Japan	0	1	1
Russia	43	0	43
Total	**64**	**10**	**74**

Out of the 10 critical attacks, six generated Snort alerts. Two rules were triggered within Snort—Unauthorized Read Request to a PLC and Unauthorized Write Request to a PLC. These rules traditionally issue alerts when an unauthorized Modbus client attempts to read or write information from a PLC or SCADA device. Both rules usually indicate that ICS network reconnaissance is occurring—the first step in ICS network exploitation.

Based on the attacks that occurred and the Snort signatures triggered, we deduced that the alerts were generated during reconnaissance as opposed to when the actual attack was carried out.

In addition to the attacks we saw, we also tracked repeat or similar IP addresses or netblocks perform attacks. One interesting statistic involved attacks against three separate honeypots that were geographically disparate. Among these attacks, we witnessed two separate /24 netblocks with five unique IP addresses performing attacks. We also witnessed referrers from Shodan queries as well as port scans, OS fingerprinting, and automated vulnerability assessments.

Many of the attacks we witnessed involved attempted exploitation of the HMI in addition to the Modbus protocol traffic. The HMI in our honeynet environment would be perceived as a gateway into the ICS environment. When the attackers attempted to modify the HMI, they were looking for SQL injection and cross-site request forgery (CSRF) vulnerabilities. SQL injection is a code injection technique that exploits security vulnerabilities in an application, often targeting the backend database. Likewise, CSRF attacks refer to a type of malicious exploitation of a website by transmitting unauthorized commands from a user that the site trusts. Attackers also often attempted to log in to secure areas using default credentials. Dictionary attacks (i.e., use brute force by nature) against an HMI were also commonly seen. As such, HMIs with no lockout mechanisms can allow attackers to attempt multiple logins with little effort and no repercussions.

Attackers who targeted Modbus traffic, meanwhile, attempted to modify and execute valid commands issued by the HMI to the PLC. Because Modbus sends traffic in cleartext without requiring authentication, it is a ripe target for attackers looking to compromise ICS environments.

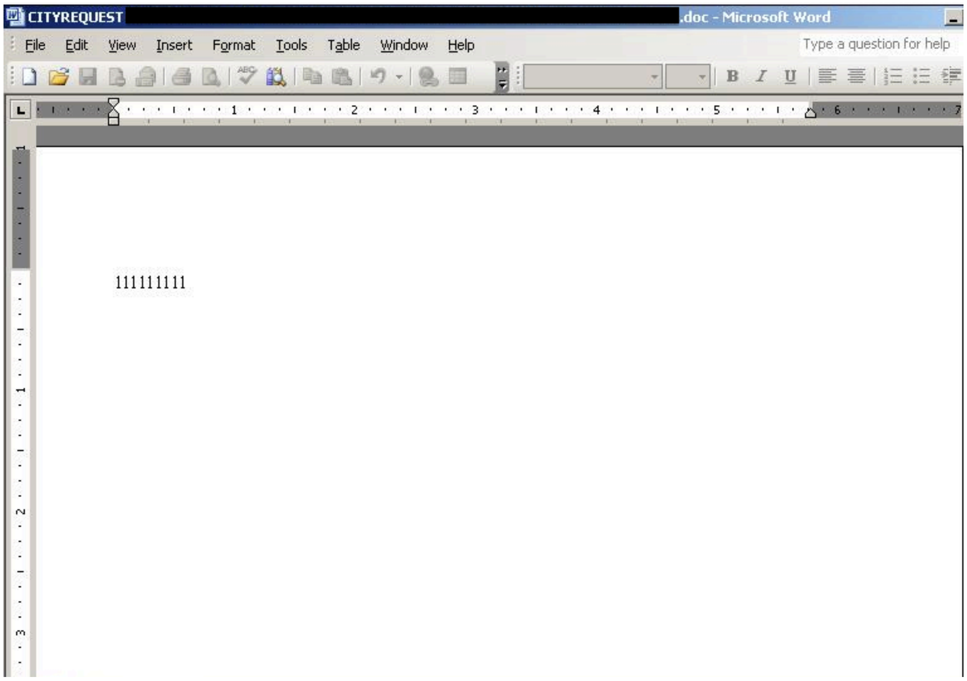
Automated Attacks

While this paper focuses on targeted attacks, we also tracked automated attacks like SQL injection attacks. The sheer number of automated attacks was surprising. For the entire honeynet during our sample timeline, we recorded 33,466 automated attacks for which 1,212 unique IP addresses were used. While we do not perform attribution or any other type of statistical analysis on these attacks, we do monitor and keep base numbers for comparison purposes.

Targeted Attacks

In the course of conducting research, we witnessed a targeted attack against a honeypot based in the United States in December 2012. Although this targeted attack took place prior to the period covered in this paper, March to June 2013, and has only been briefly discussed in "Who's Really Attacking Your ICS Equipment?," it will be discussed in greater detail here.

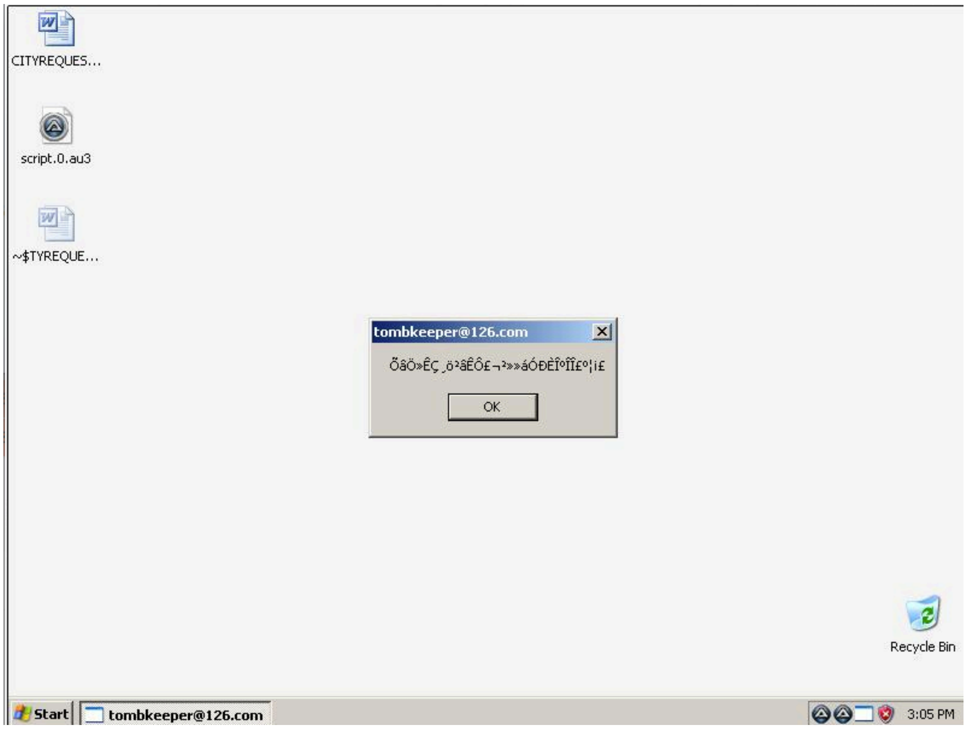
The targeted attack, like many others seen in the wild today, began with a phishing email sent to an email address provided on the website of the honeypot that was compromised. The email address was created to closely mimic a valid one that a city government would normally have. The phishing email had an attachment named *"CITYREQUEST.doc."*



111111111

Screenshot of *CITYREQUEST.doc* when opened

Opening the attached document opens a decoy document with little text defined. It also quickly and automatically closes then displays a dialog box containing unidentifiable text.



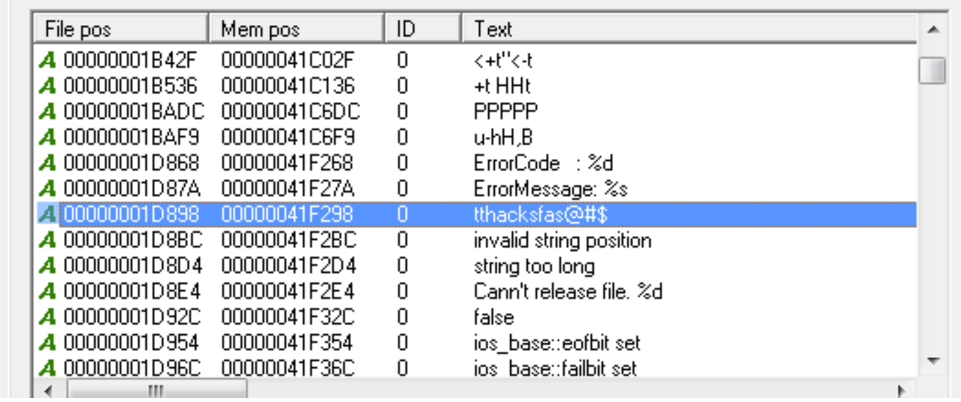
ÕâÖ»ÊÇ¸ö²âÊÔ£¬²»»áÓÐÈÎºÎÎ£º¦¡£

Dialog box that pops up after the document is close

Clicking "OK" sends out several beacons to command-and-control (C&C) servers in China and the United States. The action also leads to the dropping of two files—*ai.exe* and *gh.exe*.

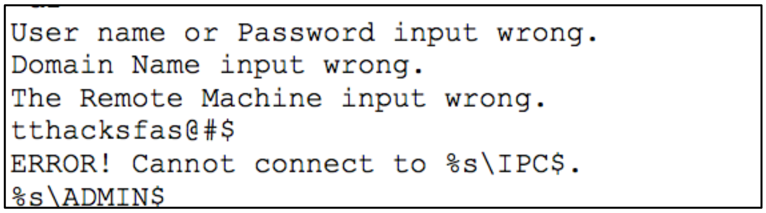
Gh.exe is a standard password hash dump file. When executed using the command line, you must run the "-w" switch to dump all of the hash's files. This is a standard functionality to maintain persistence and laterally move throughout a target network, seen in many targeted attacks.

Ai.exe, meanwhile, was more interesting. As soon as its strings were first dumped, we were quickly able to identify its origin as a common piece of malware known as “HACKSFASE.”[4]



File pos	Mem pos	ID	Text
A 00000001B42F	00000041C02F	0	<+t''<-t
A 00000001B536	00000041C136	0	+t HHt
A 00000001BADC	00000041C6DC	0	PPPPP
A 00000001BAF9	00000041C6F9	0	u-hH,B
A 00000001D868	00000041F268	0	ErrorCode : %d
A 00000001D87A	00000041F27A	0	ErrorMessage: %s
A 00000001D898	00000041F298	0	tthacksfas@#$
A 00000001D8BC	00000041F2BC	0	invalid string position
A 00000001D8D4	00000041F2D4	0	string too long
A 00000001D8E4	00000041F2E4	0	Cann't release file. %d
A 00000001D92C	00000041F32C	0	false
A 00000001D954	00000041F354	0	ios_base::eofbit set
A 00000001D96C	00000041F36C	0	ios_base::failbit set

String showing HACKSFASE



```
User name or Password input wrong.
Domain Name input wrong.
The Remote Machine input wrong.
tthacksfas@#$
ERROR! Cannot connect to %s\IPC$.
%s\ADMIN$
```

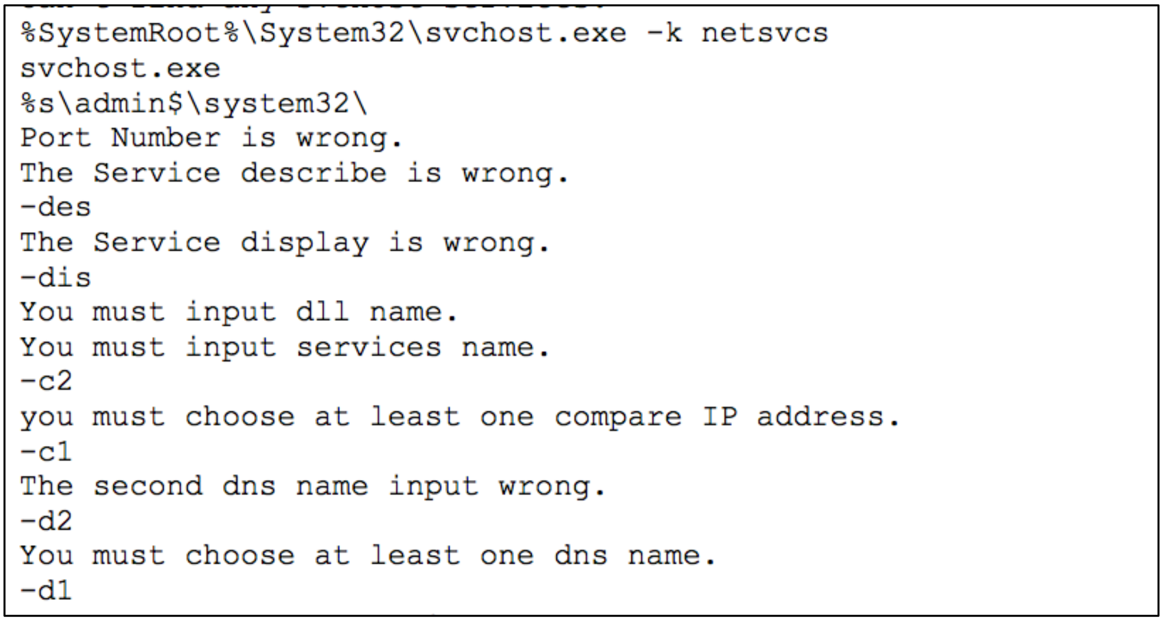
Additional HACKSFASE reference

Further analysis of *ai.exe* yielded several switches that could be used to interface with it.

```
< ai.exe –d1 (Domain) –c1 (Compare IP) –s (Service) >
```

Example of a command structure for *ai.exe*

[4] Mandiant. “APT1: Exposing One of China’s Cyber Espionage Units.” Last accessed July 4, 2013, http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf.



```
%SystemRoot%\System32\svchost.exe -k netsvcs
svchost.exe
%s\admin$\system32\
Port Number is wrong.
The Service describe is wrong.
-des
The Service display is wrong.
-dis
You must input dll name.
You must input services name.
-c2
you must choose at least one compare IP address.
-c1
The second dns name input wrong.
-d2
You must choose at least one dns name.
-d1
```

Code showing HACKFASE strings

The attackers' execution of *ai.exe* also led to data exfiltration, which began roughly three hours after *CITYREQUEST.doc* was opened. The items exfiltrated by the attackers include the Security Accounts Manager (SAM) database, VPN configuration files, and some additional configuration details like hostname, IP address, and location.

We also watched the attackers send a litany of commands via the server, many of which appeared to be for lateral movement. We noticed several "pings" and "traceroutes" to default gateways and adjoining networks. Also seen were many "arp" commands to look for communication patterns. In addition, we noticed the mounting of shared drives and folders as well as the disablement of local host-based firewalls and antivirus software. One striking item the attackers performed involved basic antiforensic techniques like deleting prefetch data on Windows® instances.

In traditional targeted attacks, these commands typically mean that the attackers are looking to maintain persistence in and laterally move throughout the target network.

Attack Statistics and Motivations

Attributing attacks is often very difficult to do. Accurately ascertaining who attacked your device is a daunting task and will only provide you a small subset of possible motivations. Determining motivations is also very difficult to do, as attackers would nearly never reveal their real intentions.

Most attack attribution attempts begin with determining the attackers' country of origin. Doing this will also help us ascertain their motivations. If Country A, for instance, is interested in copying Country B's ICS device deployment methodology, then it's possible to derive Country A's motivation behind the attack.

As shown by data from our honeynet, many of the attacks targeted deployments in Russia. It is, however, also clear that most attacks originated from the same country. In fact, roughly 58% of the total number of attacks targeted deployments in Russia. The "cannibalistic" nature of attacks can easily be confirmed by looking at the honeypot data. Each honeypot deployed within Russia used a Russian IP address. Russian IP addresses launched noncritical attacks against the Russian honeypots for a total of 43 times.

Among the critical attacks across the honeynet, five or 50% originated from China. It is interesting to note that we recorded four IP addresses from China launching attacks. These four IP addresses also resided in two /24 networks.

When attempting to determine possible motivations, we should also consider the type of attack that ensued. If an attack was targeted in nature, for instance, but didn't compromise the operation of the target ICS device, the attackers' motivation could be espionage or information gathering. If an attack, however, compromised the operation of a target ICS device, depending on how badly it was affected, then the motivation could be considered destructive in nature.

Among all of the attacks seen across the architecture from December 2012 to May 15, 2013, we can accurately say that at least 15 were targeted in nature and aimed to gather information, spy on the target, or compromise the target's operation. At least 33 attacks appeared to be destructive in nature and aimed to halt the operation of a target ICS device. These could be attacks of happenstance, wherein the attacker just happened to come across the honeypot, or targeted. But establishing the motivation behind these 20 attacks was more difficult. We did not consider "accidental" attacks in our assessment, as counting such attacks and proving they happened is difficult to do. An accidental attack can occur when an attacker with a nondestructive motive accidently causes a critical or destructive attack against a target ICS infrastructure.

We did not consider or account for attacks of happenstance as well. These attacks occur when someone searching Shodan, for instance, happens to see an external-facing ICS device and decides to attack it.[5] While many of the attacks we saw started out with Shodan queries, we cannot accurately say if these were accidental or targeted in nature without additional details like port scans.

Conclusion

We cannot accurately say how often attacks against true ICS devices occur in the wild but we can say that attacks against unprotected or semi-protected ICS devices occur in the wild on a somewhat regular basis. The findings in this paper help illustrate that the ICS device threat landscape constantly changes at a seemingly rapid pace. It also illustrates that attacks against ICS devices are occurring and simply ignoring the fact that they do will not make the problem go away.

As with any security problem, using a multitier approach is the best solution. Heeding the recommendations in "Who's Really Attacking Your ICS Equipment?" and enabling the following controls can help your organization thwart ICS attacks:

- **Implement a USB/external media lockdown:** A surprising number of ICS attacks start out from an infected USB drive. As such, do not allow the use of USB drives and provide read/write access to any external media on any ICS device.

- **Use proactive protection:** While many oppose the use of intrusion prevention system (IPS) or any sort of proactive protection on an ICS network, we believe doing so can help thwart lateral movement. Not all networks can support proactive protection though, so use this only when applicable.

- **Whitelist applications:** In any ICS environment, it is important to not only know what applications are present, it is also imperative to control what are installed. Application whitelisting alleviates a lot of the stress involved in using application control. Application whitelisting, for one, only allows approved applications to be installed on a control network. This reduces the overall likelihood of vulnerability exploitation, in addition to minimizing the amount of communication that originates from a "protected" ICS network.

[5] *SHODAN.* Last accessed, July 4, 2013, www.shodanhq.com.

- **Classify data:** Knowing what data resides in or traverses an ICS network is very important in understanding the risks losing it can pose to an environment. Classifying data into "highly confidential," "confidential," and/or "open access" types can help ensure that important and confidential documents do not make their way out of your ICS environment. Doing the same thing to information that comes in to the environment should enhance protection as well.

- **Follow a standard:** While many standards don't cover necessary topics many security experts would consider crucial, some ICS standards are very good. Following National Institute of Standards and Technology (NIST)—the U.S. government's ICS standards body—standards is a great starting point to get your ICS network in order.

- **Red team often:** While many are opposed to "red teaming" or penetration testing on networks or applications on an ICS network, research has proven that this often helps lower vulnerability counts and ensures that vulnerabilities are addressed. Performing red teaming on a quarterly basis, for instance, will help ensure that vulnerabilities are patched in a timely fashion.

- **Manage vulnerabilities:** Similar to red teaming, vulnerability management will also help ensure that vulnerabilities, especially critical ones, are patched. Introducing a vulnerability scanner and manager to your ICS infrastructure will help lower your vulnerability count and help drive awareness of the issues plaguing your ICS environment.

Appendix

The following table shows more details regarding the attack types made against which particular honeypot deployment.

Target Origin	Brazil	Russia	USA	Ireland	Singapore	China	Japan	Australia
Netherlands	N/A	2 noncritical	N/A	N/A	N/A	N/A	N/A	N/A
China	N/A	1 noncritical 3 critical	N/A	1 critical	N/A	1 critical 1 noncritical	N/A	N/A
Germany	N/A	4 noncritical 1 critical	N/A	N/A	N/A	N/A	N/A	N/A
Kazahkstan	N/A	1 noncritical	N/A	N/A	N/A	N/A	N/A	N/A
Canada	N/A	1 noncritical	N/A	N/A	N/A	N/A	N/A	N/A
USA	N/A	2 noncritical	N/A	N/A	N/A	1 noncritical	N/A	N/A
Australia	N/A	1 noncritical	N/A	N/A	N/A	N/A	N/A	N/A
Moldova	N/A	1 noncritical	N/A	N/A	N/A	N/A	N/A	N/A
Ukraine	N/A	2 noncritical	N/A	N/A	N/A	N/A	N/A	N/A
UK	N/A	N/A	N/A	N/A	N/A	1 critical	N/A	N/A
France	N/A	N/A	N/A	N/A	N/A	1 critical	N/A	N/A
Palestine	N/A	1 noncritical	N/A	N/A	N/A	1 critical	1 noncritical	N/A
Poland	N/A	1 noncritical	N/A	N/A	N/A	N/A	N/A	N/A
Slovenia	N/A	1 noncritical	N/A	N/A	N/A	N/A	N/A	N/A
Japan	N/A	1 critical	N/A	N/A	N/A	N/A	N/A	N/A
Russia	N/A	43 noncritical	N/A	N/A	N/A	N/A	N/A	N/A

Trend Micro Incorporated, a global leader in security software, strives to make the world safe for exchanging digital information. Our innovative solutions for consumers, businesses and governments provide layered content security to protect information on mobile devices, endpoints, gateways, servers and the cloud. All of our solutions are powered by cloud-based global threat intelligence, the Trend Micro™ Smart Protection Network™, and are supported by over 1,200 threat experts around the globe. For more information, visit www.trendmicro.com.

©2013 by Trend Micro, Incorporated. All rights reserved. Trend Micro and the Trend Micro t-ball logo are trademarks or registered trademarks of Trend Micro, Incorporated. All other product or company names may be trademarks or registered trademarks of their owners.

Securing Your Journey
to the Cloud

10101 N. De Anza Blvd.
Cupertino, CA 95014

U.S. toll free: 1 +800.228.5651
Phone: 1 +408.257.1500
Fax: 1 +408.257.2003