The GasPot Experiment:
Unexamined Perils in Using Gas-Tank-Monitoring Systems

Kyle Wilhoit and Stephen Hilt
Forward-Looking Threat Research (FTR) Team

A **TrendLabs**SM Research Paper

Contents

1
Electronic failures in gas-monitoring systems

3
GasPot architecture and deployment

8
Actual attacks

14
Other possible attack scenarios

17
Conclusion

SECTION I

Electronic failures in gas-monitoring systems

Electronic failures in gas-monitoring systems

An explosion rattled the sleepy town of Bayamon, Puerto Rico, in the wee hours of 23 October 2009 [1]. The fire blazed for three days, burning down houses and causing thick black clouds of gasoline-fueled smoke and forcing residents to flee their homes. The culprit behind the catastrophe? Investigators said it was a glitch in the facility's computerized monitoring system. A storage tank was getting refilled with gasoline from a fuel ship docked along the San Juan harbor. Since the tank's meter malfunctioned, the petrol kept overflowing until it met an ignition source. The burning district became the aftermath.

In places like the United States (US) and others worldwide, gas stations are primarily privately owned. Some business owners can be described as independent, tech-savvy, and modern. Gas retailers are aware of the risks tied to their business and so heavily invest in equipment that allow them to remotely monitor and manage gas levels to avoid industrial accidents. An explosion of any kind is considered dangerous. Physical damages can have an irreversible impact on a business's bottom line or the business itself, if an explosion is sufficiently large enough to deplete its assets.

For some months now, several Guardian AST gas-tank-monitoring systems have suffered electronic attacks [2], possibly instigated by hacktivist groups like Anonymous. Successful attacks can affect inventory control, data gathering, and delivery tracking, in turn impacting the availability of gasoline in local stations.

To better understand the current gas-tank-monitoring system attack landscape, we developed a way to simulate the existence of these devices to check whether threat actors will find them venues attractive enough to go after. We created virtualized Guardian AST tank-monitoring systems, complete with function and input/output (I/O) controls and other features, that make attackers believe they are real. These are essentially gas-tank-monitoring system honeypots, hence the nickname, "GasPot."

We observed the attacks and watched what the attackers did, essentially gathering intelligence on the nefarious actors. Unlike previous Trend Micro honeypot-deployment projects, which only focused on critical infrastructure [3], this research features attacks against noncritical industrial control systems (ICS). It was interesting to see if attackers consciously stayed away from more visible critical infrastructures due to legal ramifications. Attacks against noncritical devices that can't cause as much large-scale harm could serve as practice for more damaging attacks.

SECTION II

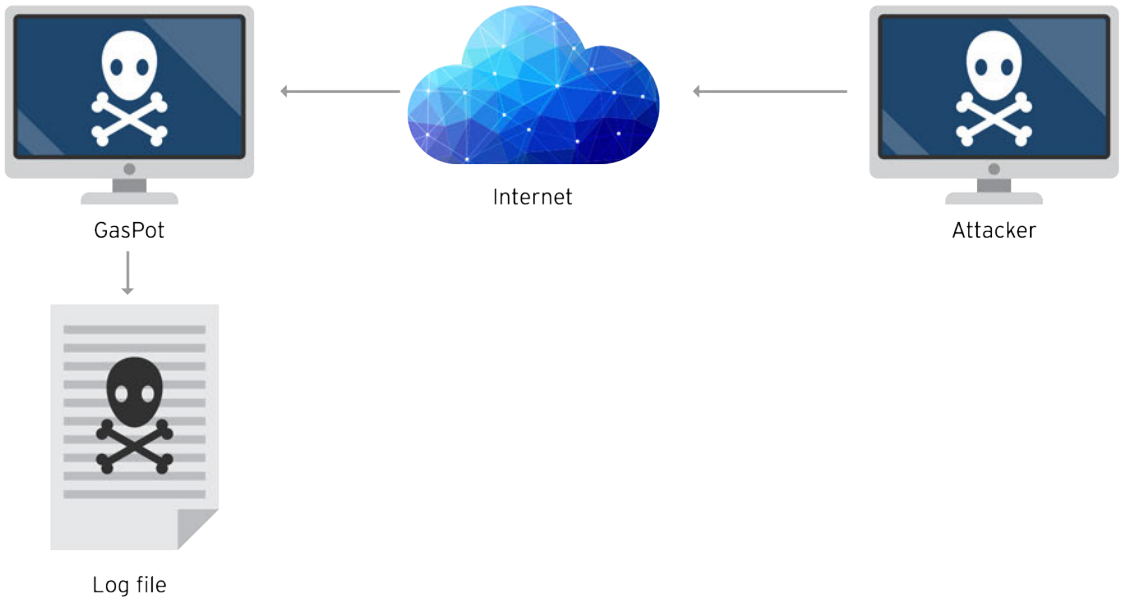
GasPot architecture and deployment

GasPot architecture and deployment

Architecture

GasPot comprises a single Python script written to serve as a functional honeypot that logs connections and compromise attempts. It was designed from the ground up to look like no other existing virtual honeypot. Each instance that it runs is unique, making honeypot fingerprinting harder for attackers to do.

GasPot has a simple architecture that gets locally logged on the device that runs it. This way, when GasPot runs, no additional services that would make it appear to be anything other than an authentic device would run. Each GasPot instance logs the time in Coordinated Universal Time (UTC), allowing easy comparison across multiple instances.



Basic architecture used in GasPot deployment, which can be achieved via router or firewall access control lists (ACLs) instead of direct Internet deployment to limit exposure to systems that collect logs

When deployed, GasPot supports six different commands, including one that allows users to change values. This has been included in part to simulate attack vectors observed in systems found on the Internet.

```
# If the response is the I20100 command, print the proper information
if "I20100" in response:
    # log it was an I20100 command
    target.write(str(datetime.datetime.utcnow().strftime('%m/%d/%Y %H:%M')) + \
        " - I20100 Command Attempt from: %s\n" % addr[0])
    conn.send(I20100())
```

Sample command entered to pull out tank information; also writes information entered, along with the Internet Protocol (IP) address of the user who entered the data to the log file

The results of this command are pseudo-randomized upon GasPot's startup, aided by Python's randomization feature within plausible ranges.

```
# Temperature of the tank, this will need to be between 50 - 60
temp1 = str(random.randint(50, 60)) + "." + str(random.randint(10, 99))
temp2 = str(random.randint(50, 60)) + "." + str(random.randint(10, 99))
temp3 = str(random.randint(50, 60)) + "." + str(random.randint(10, 99))
temp4 = str(random.randint(50, 60)) + "." + str(random.randint(10, 99))
```

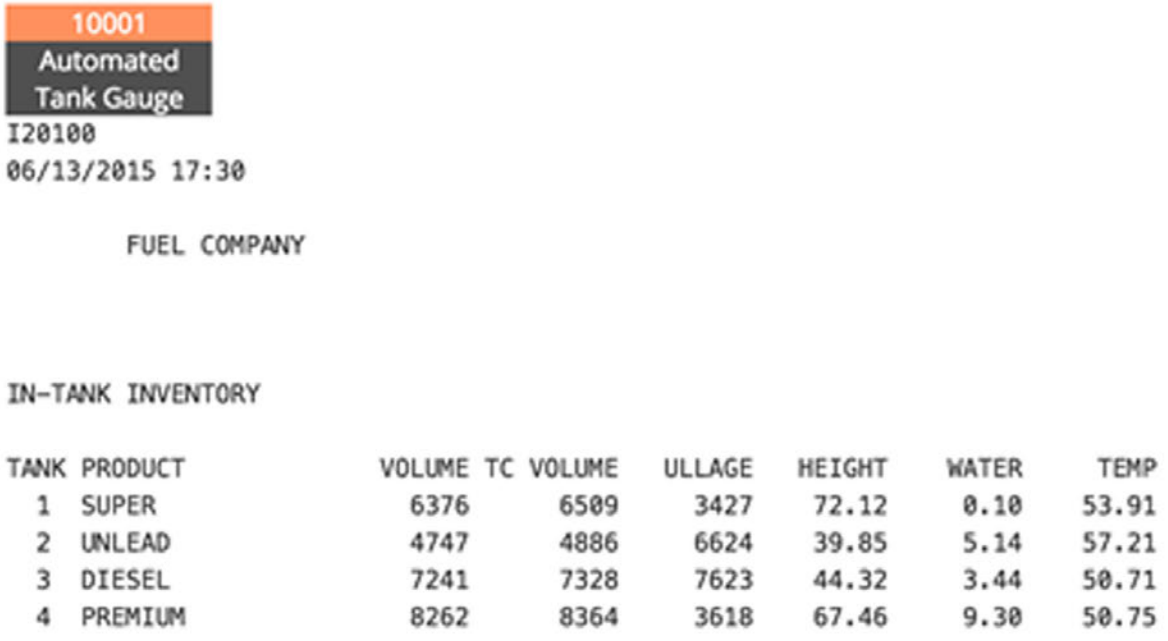
Sample results of the command above, which show specific tanks' temperatures

The temperatures (in the figure above) may be changed to suit the temperature measurement standard (Fahrenheit versus Celsius) used in the region where the device is located. GasPot users may also want to consider changing values to simulate above- or in-ground tanks.

As shown in the example, values including temperature, tank name, volume, and other information have been assigned. Attackers who enter the command will then get the values as response with the current time stamp. GasPot users can also come up with a static list of station names, the number of which may vary. Like the other values, these should be changed based on deployment location. Using local gas station names will make GasPot more realistic.

```
# This function is to set-up up the message to be sent upon a successful I20100 command being sent
# The final message is sent with a current date/time stamp inside of the main loop.
def I20100():
    I20100_1 = '''
I20100
'''
    I20100_2 = '''

    ''' + station + '''



IN-TANK INVENTORY

TANK PRODUCT             VOLUME TC VOLUME   ULLAGE   HEIGHT    WATER     TEMP
  1  '''+ PRODUCT1 + '''''' + str(Vol1) + '''      '''+ str(vol1tc) +'''     '''+ ullage1 +'''     '''+ height1 +
     '''      '''+ h2o1 +'''     '''+ temp1 +'''
  2  '''+ PRODUCT2 + '''''' + str(Vol2) + '''      '''+ str(vol2tc) +'''     '''+ ullage2 +'''     '''+ height2 +
     '''      '''+ h2o2 +'''     '''+ temp2 +'''
  3  '''+ PRODUCT3 + '''''' + str(Vol3) + '''      '''+ str(vol3tc) +'''     '''+ ullage3 +'''     '''+ height3 +
     '''      '''+ h2o3 +'''     '''+ temp3 +'''
  4  '''+ PRODUCT4 + '''''' + str(Vol4) + '''      '''+ str(vol4tc) +'''     '''+ ullage4 +'''     '''+ height4 +
     '''      '''+ h2o4 +'''     '''+ temp4 +'''
'''
    return I20100_1 + str(TIME.strftime('%m/%d/%Y %H:%M')) + I20100_2
```

Sample gas tank information entered into GasPot



Sample of an actual GasPot instance that can be found on Shodan[2], a search engine that crawls the Internet for connected devices

We tested GasPot's functionality on a variety of scanners and tools to ensure that it can be discovered and interacted with, much like an actual device.

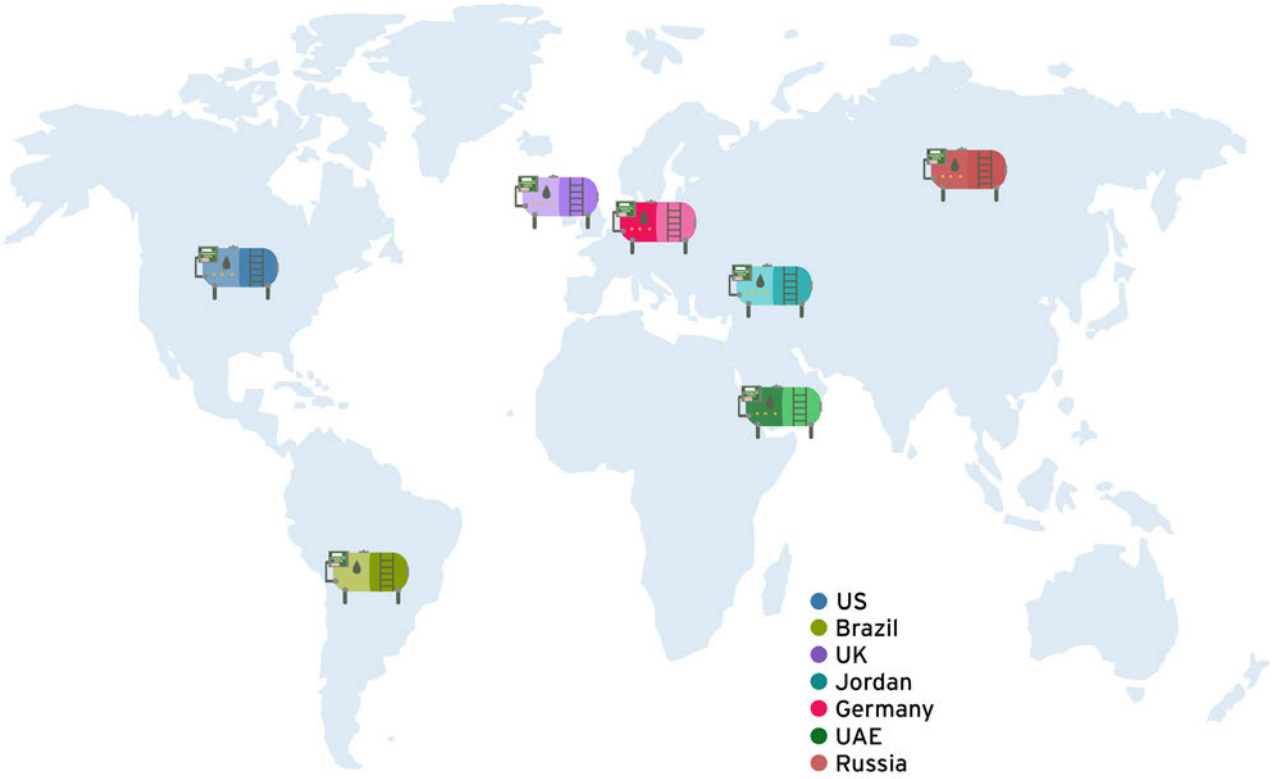
2 *http://www.Shodanhq.com/*

Deployment

To observe attacks, GasPot was deployed across regions to ensure the collection of data worldwide. We deployed more GasPot instances in the United States than in any other country to reflect reality (more publicly accessible authentic systems were found in the country than in any other). We kept the number of deployments small though so they would not stand out in any given region.

To gather data worldwide, we deployed GasPot in the following countries:

- US
- Brazil
- United Kingdom (UK)
- Jordan
- Germany
- United Arab Emirates (UAE)
- Russia



Graphically shows where GasPot was deployed (not exact)

We used various approaches to deploy GasPot instances. We configured some to appear on Shodan but left others so they would not be publicly searchable even if they are Internet connected. We did this so we can collect data on groups who potentially index automatic tank-gauging (ATG) systems without using preexisting reconnaissance tools.

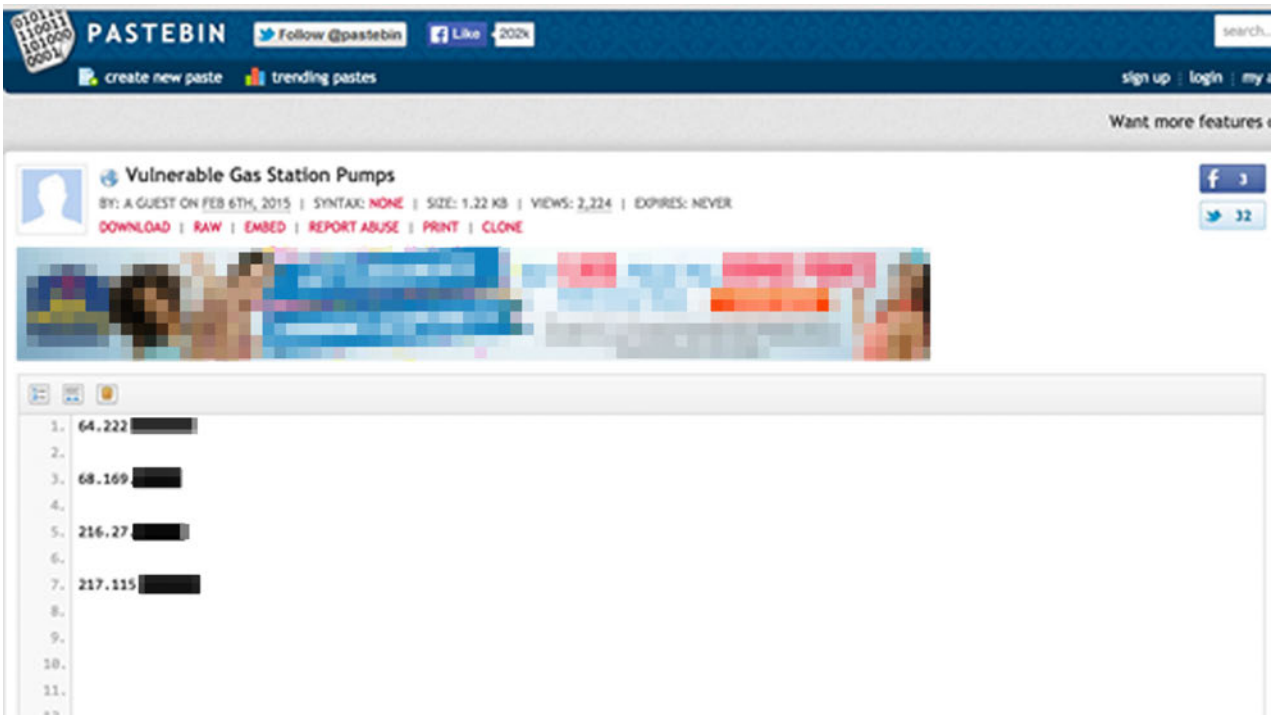
All deployments were done on physical IP addresses. No cloud service providers were used to ensure that the devices looked as real as possible.

SECTION III

Actual attacks

Actual attacks

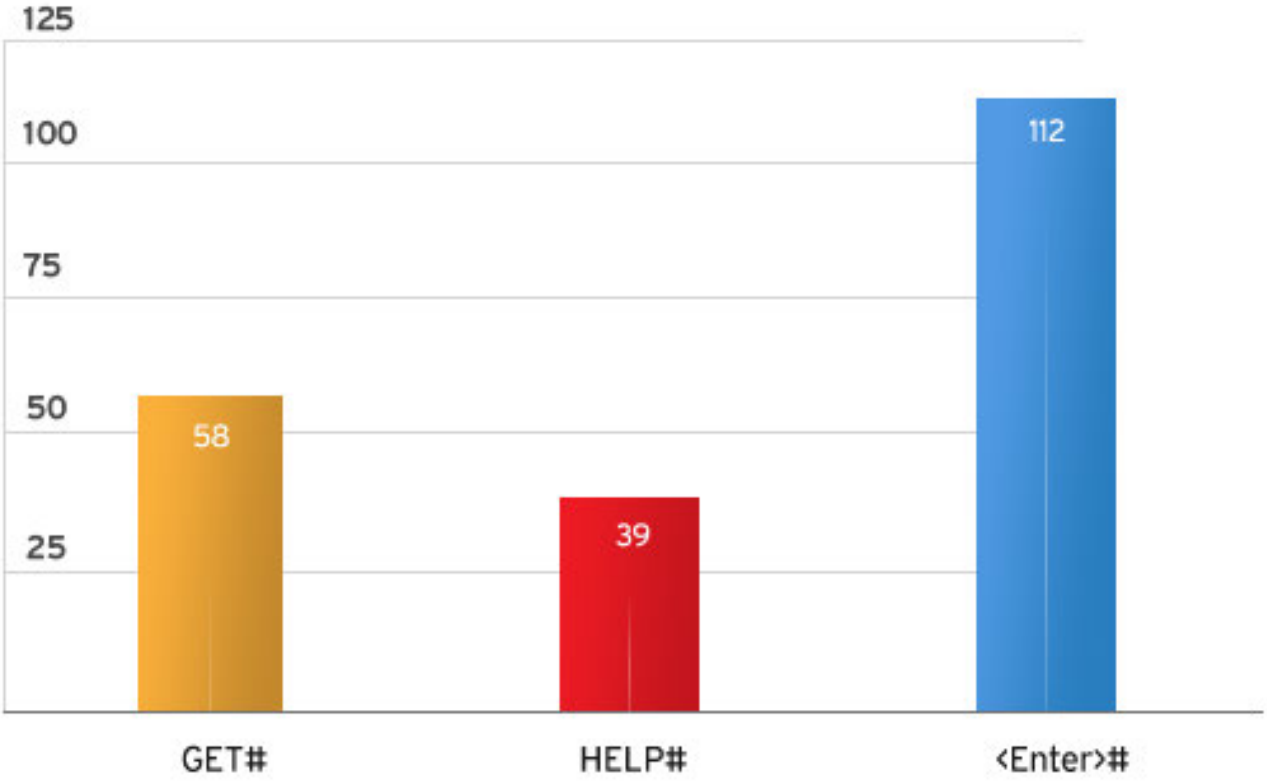
As part of our regular monitoring, we watched underground forums and other sources such as Pastebin for mentions of any GasPot deployment.



Sample Pastebin post where one GasPot instance, along with real systems, were discussed

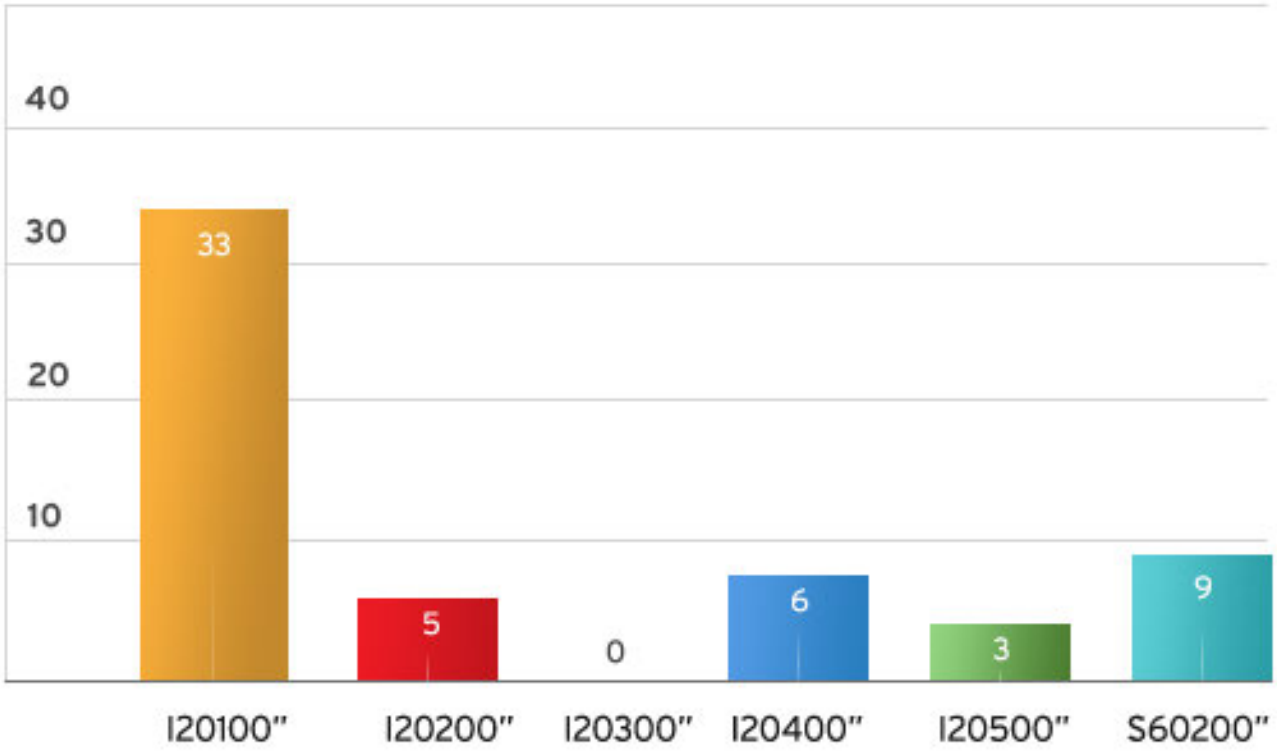
As shown (in the figure above), attackers actively looked for and shared information on ATG systems, including instances of GasPot, discovered online. Interestingly, however, none of the links discussed called out instances of GasPot as honeypots. This cemented the fact that we achieved our goal—to make GasPot appear as real as possible.

Regularly monitoring GasPot allowed us to observe reconnaissance done on our system. Most of the connections we observed can be categorized as done by automated scanners performing <#> GET and <#> HELP command requests as well as <#> carriage returns.



Breakdown of attempts seen on GasPot categorized as "automated-scanning" and "basic-connection" attempts

We also tracked the number of valid commands that were entered into GasPot. We broke these down to only those that were fully implemented on GasPot. The command most commonly entered was "I20100" (33 times), which lists basic tank information. Existing Nmap scripts pull out this information from GasPot. Shodan also uses this command to gather information about systems.



Various commands used on GasPot

A command that allows users to make changes to systems—"S60200"—was also used on GasPot. This was used nine times to change GasPot's tank names. Incidentally, this command was used on real systems in the past. One such instance was when the name of a tank in Maine was changed to "WE_ARE_LEGION" [4].

Our monitoring of GasPot deployments revealed that several were modified using the commands, "S60201" and "S60203," which changed the names of tanks 1 and 3, respectively. We first spotted modifications on tank information on a GasPot deployed in Jordan.

```
06/27/2015 08:47- Connection from : 5.106.221.208
06/27/2015 08:47 - S60201: H4CK3D by IDC-TEAM Command Attempt from: 5.106.221.208
```

Log file showing the first modification observed on tank 1 of a system

Tank 1's name (in the figure above) was changed to "H4CK3D by IDC-TEAM." IDC-TEAM , also known as the "Iranian Dark Coders Team,"[3] is a group of security enthusiasts operating in Iran. It is a pro-Iran group responsible for website defacements, information sharing, malware distribution, and hacktivisim. The group usually uses the phrase, "H4CK3D by IDC-TEAM," in website defacements.

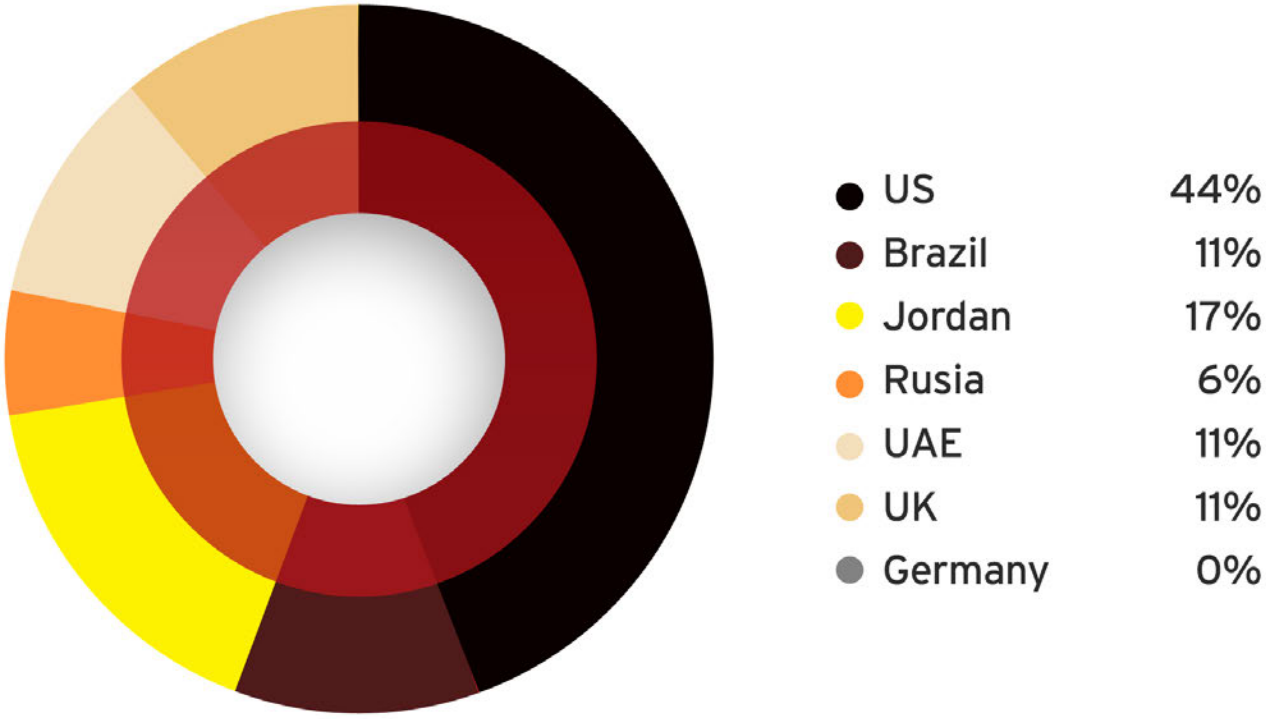
3 *http://idc-team.net/*

```
06/27/2015 08:50- Connection from : 2.147.147.123
06/27/2015 08:50 - S60203: AHAAD WAS HERE Command Attempt from: 2.147.147.123
```

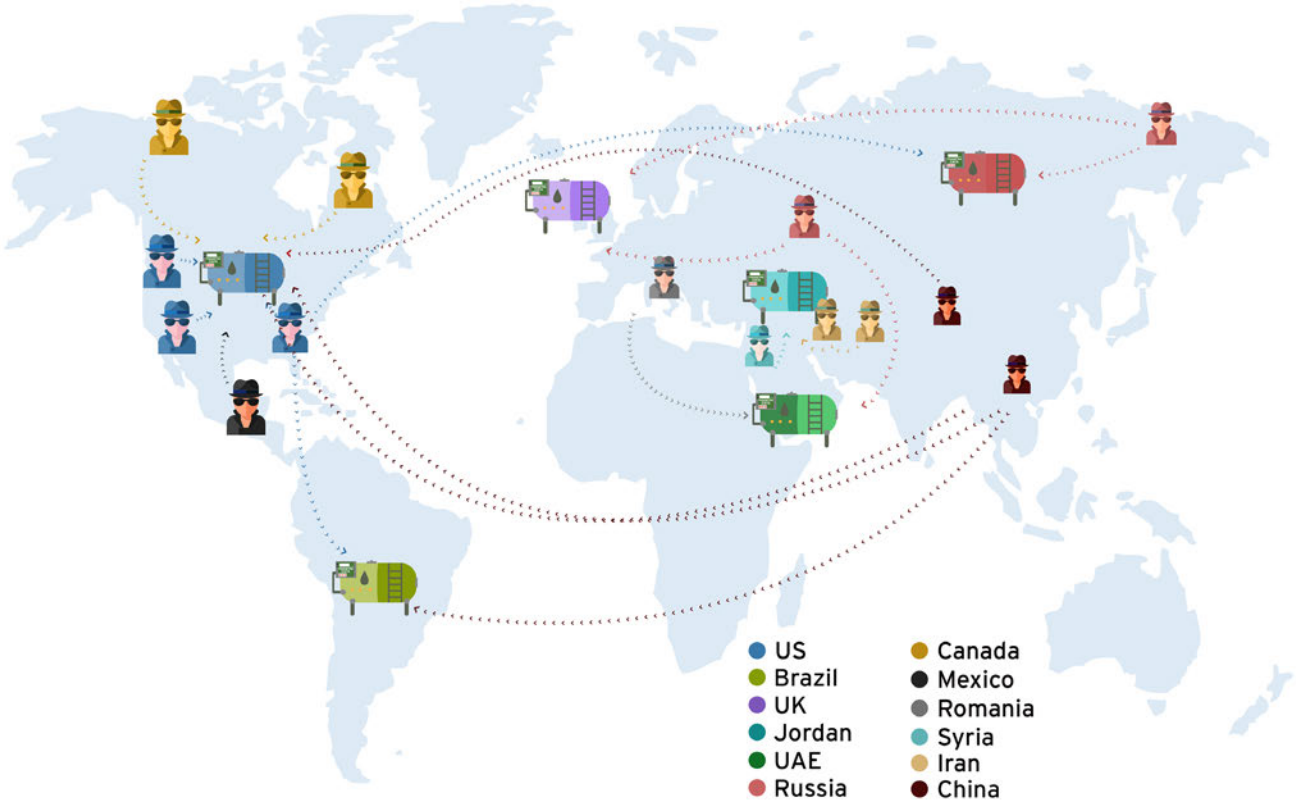
Log file showing modifications done on another GasPot in Jordan

Another GasPot (in the figure above) was modified to say "AHAAD WAS HERE" by the same group or person behind the first attack. AHAAD is an alias mentioned in some website defacements performed by IDC. Note, however, that both modifications could have been performed by another group or person who wishes to make them look like deeds by IDC.

We found a total of four modifications made on GasPot deployments.



Breakdown of attacks on GasPot deployments observed by country



Attacks seen on GasPot deployments
Note: *The attackers can use proxies or virtual private networks (VPNs), which may skew the origin data.*

The GasPot systems deployed in the United States were most attacked. The United States accounted for 44% of the total number of attacks followed by Jordan (17%). Interestingly, the GasPot deployments in Germany did not suffer attacks.

One of the attacks against a US-based GasPot was not against the system itself. It was a distributed denial-of-service (DDoS) attack against a GasPot instance for a period of two days. At its height, the attack was roughly around 2Gbps and appeared to be a Low-Orbit Ion Cannon (LOIC)-tool-based DDoS attack [5]. It was observed on a GasPot deployed in the Washington D.C. area. Based on evidence, it was believed to have been caused by the Syrian Electronic Army (SEA) [6].

```
GET /app/?id=17783745&msg=SEAcannnGO HTTP/1.1Host:OBFUSCATED-Agent:
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.2.12) Gecko/20101026
Firefox/3.6.12Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8Accept-Language:
en-us,en;q=0.5Accept-Encoding: gzip,deflateAccept
```

Shows the message of the GET command, "SEAcannnGO," in the attack mentioned above

We have not seen the SEA use this technique in previous attacks. It could have been performed by another group or person who wants to put the blame on the SEA.

SECTION IV

Other possible attack scenarios

Other possible attack scenarios

Why would attackers target electronic gas-tank-monitoring systems? Motivations vary for different types of threat actors.

Pranks

Hackers can simply be testing out their skills against ATG systems, experimenting and checking what level of access they can get and what they can do with it. Pranksters can, for instance, change the tank labels to something menacing.

Reconnaissance

Threat actors can use the information visible on Internet-facing ATG systems to perform preliminary reconnaissance for highly industry-specific targeted attack campaigns.

As shown earlier, determined attackers can learn the following information about a certain facility or gas station—kinds of gas tanks monitored by a system, gas levels in each tank, existence of leaks, and current status of tanks.

Attackers can use the information to determine when a facility may be expecting the next delivery. They can then, for instance, craft a social engineering attack leveraging the information and eventually get inside the system's network for further reconnaissance.

Extortion

Extortion is especially prevalent in the cybercriminal landscape. Should "set console password" be a feature in ATG systems, attackers can reset a password, especially if the default is still in use, in order to lock system owners out. They can then hold the console hostage and ask for ransom to restore owner access.

Attackers can show off what else they can do once they gain control of a console. They can say that they have switched label names so that, for instance, one tank is no longer labeled "Unleaded" but "Premium," which can lead to some operational issues.

Small-scale sabotage

While attacks against ATG systems will in no way immobilize an entire nation, their weaknesses will definitely present unique opportunities for attackers. Earlier, we talked about what can happen if monitoring systems fail and seen real-world attacks that modified parameters in ATG systems. Other commands in similar systems can not only modify tank labels, but also tank levels and overflow limits, temperature compensation values, tank tilt and diameter values, and other units of measurement.

Given certain conditions, attackers can, for instance, set a tank overflow limit to a value beyond its capacity, thus triggering an overflow. And as shown earlier, gas overflows are extremely dangerous because the liquids they contain are highly combustible.

Conclusion

Conclusion

Searches using Shodan reveal the sheer number of Internet-connected devices worldwide. These include a smorgasbord of not only personal and home Internet of Things (IoT) devices like routers, baby monitors, and heating systems, but also surveillance cameras, traffic lights, medical equipment, and power plants.

Traditionally, supervisory control and data acquisition (SCADA) and ICS should not be Internet connected unless absolutely necessary. If they really need to be, their security should be so strong that access to them is extremely limited and private. What is frightening though is that, time and again, a scan of sites like Shodan expose systems that shouldn't even be Internet connected. Even worse, these devices use the barest security barriers, if at all. And so, anyone with enough time and motivation can leverage access to these for pranks, reconnaissance, extortion attacks, or even small-scale sabotage.

As shown, attacks against Internet-facing gas-tank-monitoring systems are no longer hypothetical. In the course of doing research, we found existing attacks on Guardian AST gas-tank-monitoring systems, and not only against our GasPot deployments.

We found that GasPot systems deployed in the US were deemed most attractive by attackers. In fact, 44% of the attacks we saw targeted these, followed far behind by Jordan (17%). GasPots in Brazil, the UK, the UAE, and Russia were also attacked. GasPots deployed in Germany, however, were not. All these showed an ongoing interest in accessing and attacking Internet-facing ATG systems, and that this interest is somewhat also prevalent outside the US. We also found that apart from ATG systems, users of underground forums and hacker collectives showed an interest in SCADA systems.

On a broader scale, the implications of this research highlight the lack of security awareness surrounding Internet-connected devices. We would like the conversation to revolve around unsecured SCADA devices, of which ATG systems comprise only one example. Vendors of these devices should become accountable for the security weaknesses of both the devices they offer and the OSs used to manage them. Security should be built from the ground up.

Gas-tank-monitoring system owners should be aware of the issues associated with hooking up ATG systems to the Internet. If they considered all options and decided that the convenience of remote access to ATG system consoles is worth the risks shown in this paper, then they should consult an information technology (IT) security expert for the sole purpose of ensuring that their connection security is sufficient.


EURO DIESEL BS
EURO DIESEL BS
EURO SUPER BS 95
4
PLEASE PAY THE BILL BEFORE STARTING YOUR VEHICLE
BITTE BEZAHLEN SIE DIE RECHNUNG VOR DEN STARTEN DES KRAFTFAHRZEUGES
PKW
LKW
EURO DIESEL BS
EURO DIESEL BS
EURO SUPER BS 95

Appendix

Appendix

Commands available on GasPot

The list of commands below are based on the manual[4]. "TT" refers to the tank number (TT = Tank Number [Decimal, 00=all]).

- **I201TT:** Stands for "In-Tank Inventory Report." This includes the product name, tank volume, tank ullage, tank height, tank's water content, and tank temperature.
- **I202TT:** Stands for "In-Tank Delivery Report." This shows the date and time of the last delivery as well as some of the same information from the I20100 command such as water, temperature, and height of tank delivered.
- **I203TT:** Stands for "In-Tank Leak Detect Report." This shows if there are any detected leaks in any of the tanks.
- **I204TT:** Stands for "In-Tank Shift Inventory Report." This is done by an attendant at a gas station to see how much gasoline, diesel, or other products have been removed from tanks during a shift.
- **I205TT:** Stands for "In-Tank Status Report." This shows the current status of tanks.
- **S602TT:** Stands for "Set Tank Product Label." The most commonly used command in attacks seen.

Entering any other command results in error messages. An example would be "9999FF1B" where "9999" means the system did not understand a command while "FF1B" is the appropriate checksum for the preceding "9999" string.

4 *http://www.veeder.com/gold/download.cfm?doc_id=7323*

References

1. Arthur Brice. (17 November 2009). *CNN*. "Puerto Rico Fire Linked to Faulty Gas-Tank-Monitoring System." Last accessed on 20 July 2015, http://edition.cnn.com/2009/US/11/17/puerto.rico.fire.investigation/index.html.

2. Kyle Wilhoit. (10 February 2015). *TrendLabs Security Intelligence Blog*. "Is Anonymous Attacking Internet-Exposed Gas-Pump-Monitoring Systems in the U.S.?" Last accessed on 9 July 2015, http://blog.trendmicro.com/trendlabs-security-intelligence/is-anonymous-attacking-internet-exposed-gas-pump-monitoring-systems-in-the-us/.

3. Kyle Wilhoit. (2013). *Trend Micro Security Intelligence*. "The SCADA That Didn't Cry Wolf: Who's Really Attacking Your ICS Equipment? (Part 2)." Last accessed on 10 July 2015, http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-the-scada-that-didnt-cry-wolf.pdf.

4. Trend Micro Incorporated. (13 February 2015). *Trend Micro Security News*. "Tampered U.S. Gas Pumps Point to Anonymous Group." Last accessed on 9 July 2015, http://www.trendmicro.com/vinfo/us/security/news/internet-of-things/tampered-us-gas-pumps-point-to-anonymous-group.

5. Infosec Institute. "LOIC (Low Orbit Ion Cannon)—DOS Attacking Tool." Last accessed on 10 July 2015, http://resources.infosecinstitute.com/loic-dos-attacking-tool/.

6. Sarah Fowler. (25 April 2013). *BBC News*. "Who Is the Syrian Electronic Army?" Last accessed on 10 July 2015, http://www.bbc.com/news/world-middle-east-22287326.

Created by:
TrendLabs
The Global Techincal Support and R&D Center of **TREND MICRO**

TREND MICRO™

Trend Micro Incorporated, a global cloud security leader, creates a world safe for exchanging digital information with its Internet content security and threat management solutions for businesses and consumers. A pioneer in server security with over 20 years experience, we deliver topranked client, server, and cloud-based security that fits our customers' and partners' needs; stops new threats faster; and protects data in physical, virtualized, and cloud environments. Powered by the Trend Micro™ Smart Protection Network™ infrastructure, our industry-leading cloud-computing security technology, products and services stop threats where they emerge, on the Internet, and are supported by 1,000+ threat intelligence experts around the globe. For additional information, visit **www.trendmicro.com**.

TREND MICRO™

Securing Your Journey
to the Cloud