



A Framework For Fingerprinting ICS Honey Pots


Mohammad-Reza Zamiri

@d3c0der



\$ whoami



-  @d3c0der
- Mohammad Reza Zamiri aka d3c0der
- Security researcher at ZDResearch
- Formerly at Iranian Central Bank CERT
- <http://scadapot.com>

Outline



1. An overview of Honeypots and ICS Honeypots
2. ICS Honeypot fingerprinting methods
3. The Framework
4. Using the framework to find Gaspots online
5. Conclusion

Honeypot?



- Computers masquerading as vulnerable
 - Recording all interactions with users
- Two broad categories, based on service and interaction level
 - High-interaction
 - Low-interaction

Industrial Control System (ICS)



- Monitors & controls the operation of devices in industrial environments
- ICS were traditionally air-gapped
 - i.e. physically isolated from the outside world
- Now linked to the Internet to allow remote control and monitoring



ICS honeypot



- A significant number of threats are directed towards ICS nowadays
 - Due to its direct physical impact on the world
- ICS Honeypots are rare, because they need to model industrial systems that are not as publicly available
 - But they are a great source of discovering attacks against ICS

Why fingerprinting?



- Cyber Threat Intelligence (CTI) services are becoming popular
- Large data used by these services is obtained via OSINT
 - The other part via proprietary sensors of the services
- If the attackers recognize these sensors, they will send wrong data to them and protect against detection
- The stealthiness of a honeypot is also an important factor in an organization's overall security strategy

How are bad guys attacking ICS



- Many ICS attacks start with scanning ICS related ports
 - Also google hacking is useful
- There are engines like **shodan** and **censys** that make searching for Internet-connected devices easy
- For a huge number of attackers finding an open ICS port is enough of an starting point
- E.g., <https://github.com/d3coder/ICS-Hunter>

Classifying Fingerprinting Methods

Looking for default configuration



- Default options for a honeypot are generally the biggest weakness
- Works well on unconfigured & misconfigured honeypots
- The bad news is that there are many ICS honeypots out there with default config!



Identifying the environment



- Scanning network services and checking operating system default open ports
- Looking for hosting services (e.g., cloud)
- OS detection with NMAP, Xprobe2, p0f, etc.
- Other related characteristics (TCP/IP headers, ICMP echo response time, etc.)

Incomplete implementation of a protocol



- Low interaction honeypots do not implement complete feature sets
- Industrial network protocols have unique features
 - For example many of them do not support encryption or even authentication
- Attackers can start to explore more features of an ICS service and investigate suspicious cases

Unusual ICS behaviors



- ICS are designed to monitor metrics such as temperature, pressure, etc.
- The result of a natural metric monitoring must be a dynamic value
- A system that demonstrates fixed/inflexible metrics is emitting unusual ICS behavior
- Just an open ICS port doesn't mean a real ICS device

Fingerprinting ICS Honeypots

Common ICS honeypots



- **Conpot** - ICS honeypot for collecting adversary motives and methods
- **GasPot** - honeypot designed to simulate a Veeder-Root Guardian AST
- **Scada-honeynet** - simulates a variety of industrial networks and devices
- **Gridpot** - Open source tools for realistic-behaving electric grid honeynets

Default config detection example / Conpot



- Previously some default signatures of Conpot were published by other researchers
- We identified some less-known signatures by investigating Conpot's configuration files

Conpot default config detection (well-known signatures)



Protocol	port	signature	Shodan	Censys
Siemens S7	102	PLC name: Technodrome	214	185
		Plant identification: Mouser Factory	215	162
		Serial number of module: 88111222	182	92

Conpot default config detection

(less-known signatures)



Protocol	Port	Signature	Shodan	Censys
HTTP	80	Last-Modified: Tue, 19 May 1993 09:00:00 GMT	240	133
TELNET	50100	Connected to [00:13:EA:00:00:0	31	-
IEC104	2404	Data Received: 680e00000000	13	-
Ethernet IP	44818	Product name: 1756-L61/B LOGIX5561	83	-

Checking a less-known signature



- Shodan saying it's an industrial control system!
- <https://www.shodan.io/host/104.250.108.68>
- Lets take a look at it's http response headers

The screenshot shows the Shodan search results for the IP address 104.250.108.68. The main information includes the IP address, a 'View Raw Data' link, and a red badge indicating it is an 'Industrial Control System'. Below this is a table of metadata: City (Palo Alto), Country (United States), Organization (DigitalFyre Internet Solutions, LLC.), ISP (DigitalFyre Internet Solutions, LLC.), and Last Update (2018-06-23T05:34:52.040019). To the right, there are sections for 'Ports' and 'Services'. The 'Ports' section shows two open ports: 80 and 44818. The 'Services' section shows that port 80 is open for TCP and HTTP. Below the services, there is a green bar with a refresh icon and the text 'HTTP/1.1 200 OK', followed by the response headers: Date: Tue, 12 Jun 2018 16:35:29 GMT; Last-Modified: Tue, 19 May 1993 09:00:00 GMT; Content-Type: text/html; Set-cookie: path=/; Content-Length: 620.

City	Palo Alto
Country	United States
Organization	DigitalFyre Internet Solutions, LLC.
ISP	DigitalFyre Internet Solutions, LLC.
Last Update	2018-06-23T05:34:52.040019

Ports

80	44818
----	-------

Services

80	tcp	http
----	-----	------

HTTP/1.1 200 OK
Date: Tue, 12 Jun 2018 16:35:29 GMT
Last-Modified: Tue, 19 May 1993 09:00:00 GMT
Content-Type: text/html
Set-cookie: path=/
Content-Length: 620

Checking a less-known signature



```
<headers>  
  <entity name="Last-Modified">Tue, 19 May 1993 09:00:00 GMT</entity>  
  <entity name="Content-Type">text/html</entity>  
  <entity name="Set-cookie">path=/</entity>  
</headers>
```

80
tcp
http



```
HTTP/1.1 200 OK  
Date: Tue, 12 Jun 2018 16:35:29 GMT  
Last-Modified: Tue, 19 May 1993 09:00:00 GMT  
Content-Type: text/html  
Set-cookie: path=/  
Content-Length: 620
```



Identifying the environment example / Conpot



- OS detection is a good way to start
- A network scan can reveal some open ports that aren't related to ICS protocols
- Here is the result for a Conpot hosted on Debian

Command: `nmap -p 1-65535 -T4 -A -v 107.181.166.76`

OS	Host	Port	Protocol	State	Service	Version
	107.181.166.76	53	tcp	open	domain	ISC BIND 9.8.1-P1
	107.181.166.76	80	tcp	open	http	Apache httpd 2.2.22 ((Ubuntu))
	107.181.166.76	102	tcp	open	iso-tsap	
	107.181.166.76	502	tcp	open	asa-appl-proto	
	107.181.166.76	554	tcp	open	rtsp	
	107.181.166.76	1723	tcp	open	tcpwrapped	
	107.181.166.76	2222	tcp	open	ssh	OpenSSH 5.1p1 Debian 5 (protocol 2.0)
	107.181.166.76	2323	tcp	open	3d-nfsd	
	107.181.166.76	4475	tcp	filtered		
	107.181.166.76	5060	tcp	open	sip	
	107.181.166.76	5555	tcp	filtered	freeciv	
	107.181.166.76	7547	tcp	filtered		
	107.181.166.76	22222	tcp	open	ssh	OpenSSH 5.9p1 Debian 5ubuntu1.9 (Ubuntu Lin



Incomplete protocol implementation example / Conpot

- In many cases (default config cases) the result of scanning Modbus on a Conpot with PLCScan is: **unknown protocol**

```
C:\Users\d3coder\Desktop\plcscan-master>plcscan.py 175.96.80.237
Scan start...
175.96.80.237:102 S7comm (src_tsap=0x100, dst_tsap=0x102)
Module : v.0.0 (0000000000000000)
Name of the PLC : outlet (6F75746c6574000)
Name of the module : Siemens, SIMATIC, S7-300 (5369656d656e732c2053494d415449432c2053372d3330300000000000000000)
Plant identification : Power Corporation (506f77657220436f72706f72617469666e00000000000000000000000000000000)
Copyright : Original Siemens Equipment (4f726967696e616c205369656d656e732045717569706d656e7400000000000000)
Serial number of module : 16111663 (31363131313636330000000000000000000000000000000000000000000000)
Module type name : IM151-8 PN/DP CPU (494d3135312d3820504e2f445020435055000000000000000000000000000000)
OEM ID of a module : (0000000000000000)
Location designation of a module: (00000000)
175.96.80.237:502 unknown protocol
Scan complete
```


Incomplete protocol implementation example/Scada-honeynet



- Source: Digitalbond

```
sjhilt@db-assessment:~/Desktop/plcscan-read-only$ python plcscan.py [REDACTED]
Scan start...
[REDACTED]:502 [Errno 104] Connection reset by peer
[REDACTED]:502 unknown protocol
Scan complete
```

Incomplete protocol implementation example/Gaspot



- Gaspot only supports five ATG display format commands
- The response to other command is a hard-coded value :
`conn.send("9999FF1B\n")`

A screenshot of a Telnet session window titled "Telnet 146.185.158.34". The window shows the text "9999FF1B" on the first line, followed by a cursor on the second line. The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner.

```
Telnet 146.185.158.34
9999FF1B
_
```


Unusual ICS behaviours example/Gaspot



- Monitoring a protocol and waiting for changes is a nice idea (no change has a bad meaning !)
- We can do it for every ICS protocol that is providing a physical quantity such as temperature, pressure , etc

Unusual ICS behaviours example/Gasport



- First check

```
Telnet 52.39.87.62

I20100
01/30/2017 00:23

STATOIL STATION

IN-TANK INVENTORY

VOLUME TC UOLUME ULLAGE HEIGHT WATER TEMP TANK PRODUCT
7433 7435 3724 43.38 1.89 57.99 1 SUPER
2879 3003 7701 51.11 3.01 59.68 2 UNLEAD
841 5890 7422 44.98 2.98 54.65 3 DIESEL 5
2192 7422 68.98 6.52 57.39 4 PREMIUM 2139

-
```


Unusual ICS behaviours example/Gasport



- After 13 hours there is no change in ullage,height,water and temp!

```
Telnet 52.39.87.62

I20100
01/30/2017 13:34

STATOIL STATION

IN-TANK INVENTORY

VOLUME TC VOLUME ULLAGE HEIGHT WATER TEMP TANK PRODUCT
7433 7610 3724 43.38 1.89 57.99 1 SUPER
2879 3009 7701 51.11 3.01 59.68 2 UNLEAD
841 5951 7422 44.98 2.98 54.65 3 DIESEL
2254 7422 68.98 6.52 57.39 4 PREMIUM 2139
```

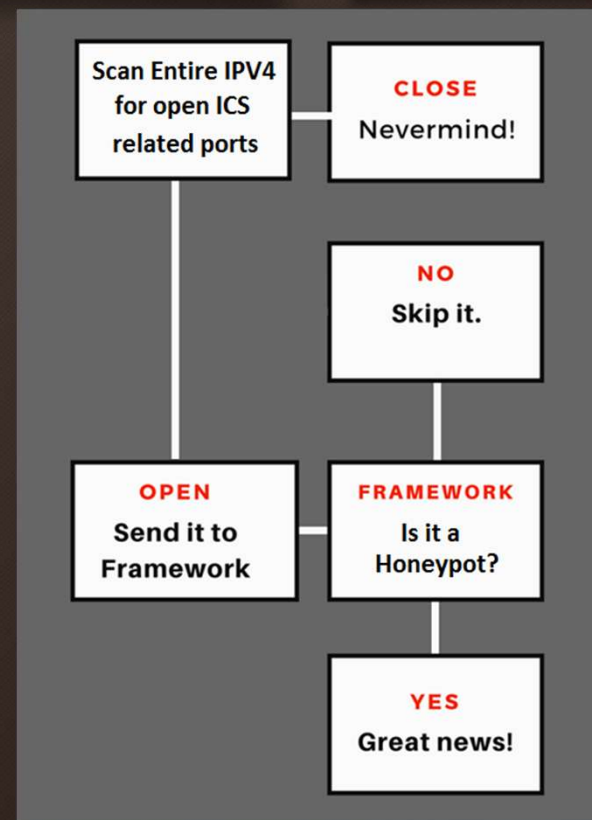


Let's run the framework!

Our Methodology



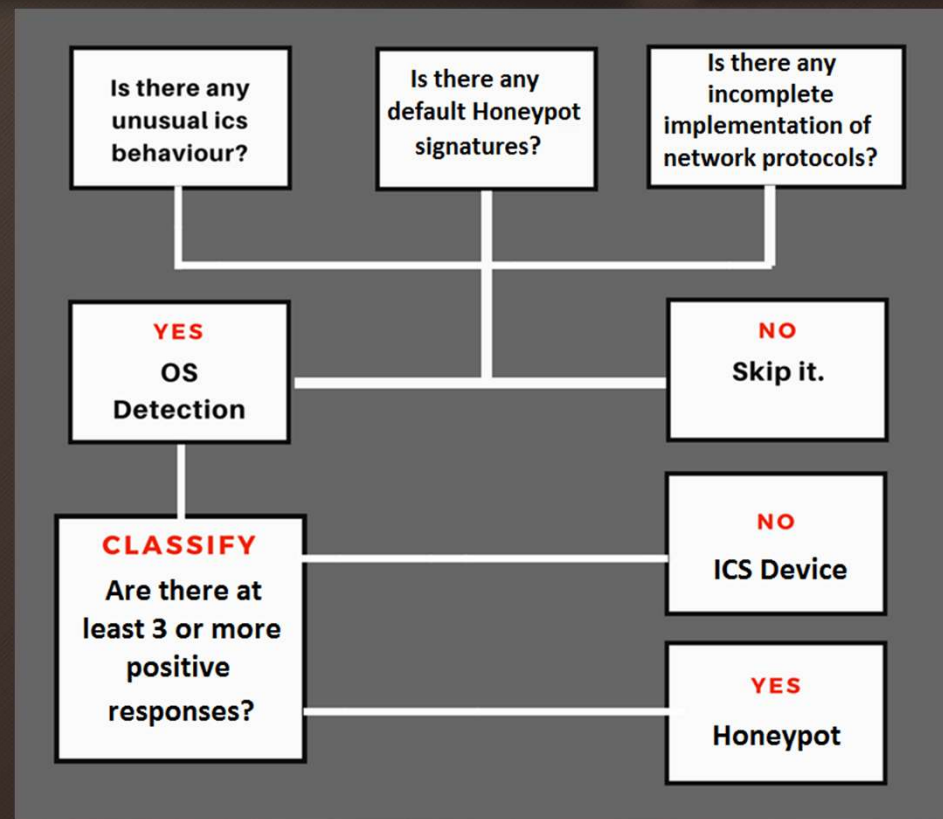
- So based on the methods we presented, We tried to implement these methods in a framework
- First we scan the whole internet by using Masscan for specified ics ports
- Then we apply our methods on the detected Ip's by using our framework



The Framework



- The framework is now available on github as part of OWASP-Nettacker project:
- https://github.com/zdresearch/OWASP-Nettacker/tree/master/lib/payload/scanner/ics_honeypot



Detecting Gaspot



- So let's detect how many Gaspot are running around the world?



Customizing the Framework



❑ Default Configuration check:

- We used the Gaspot configuration file in order to identify the default cases

❑ Incomplete protocol implementation:

- We used the "I30100" command, which was not supported by Gaspot

❑ Unusual ICS behavior:

- With a time interval of less than 2 hours, we sent two requests with I20100 command to these systems and then compared the results, so we have a change percentage

❑ OS detection :

- We used `nmap` to detect operating systems for every host with at least one positive answer to our three previous methods

Gaspot based machines analysis result



HOST:	CHANGE PERCENTAGE:	DEFAULT CONFIG:	I30100 TRAP:	NMAP OS Detection:
139.59.XX. XX	10.4166666667%	TRUE	TRUE	Linux 3.X 4.X
207.154.XX. XX	10.4166666667%	TRUE	TRUE	Linux 3.X 4.X
107.170.XX. XX	10.4166666667%	TRUE	TRUE	Linux 3.X 4.X
138.197.XX. XX	10.4166666667%	TRUE	TRUE	Linux 3.X 4.X

Real ATG device result examples



HOST:	CHANGE PERCENTAGE:	DEFAULT CONFIG:	I30100 TRAP:	NMAP OS Detection:
108.58.XX. XX	15.9090909091%	FALSE	TRUE	Larus 54580 NTP server (97%)
67.158.XX. XX	18.4210526316%	TRUE	FALSE	dell embedded (97%)
24.39.XX. XX	24.4444444444%	FALSE	TRUE	Lantronix embedded (98%)
24.250.XX. XX	32.5%	TRUE	FALSE	Linux 2.6.XOS

The final results



Number of IPv4 addresses:	Host with open 10001 ports:	ATG devices:	suspicious cases:	Gaspots:
4,294,967,296	4,133,186	4,838	102	17

How about shodan?



- There was only “9” identified Gaspot on shodan at the time of our scan

The screenshot shows the Shodan search interface. At the top, there are navigation links for 'Shodan', 'Developers', 'Book', and 'View All...'. The search bar contains 'product.gaspot'. Below the search bar, there are buttons for 'Exploits', 'Maps', 'Share Search', and 'Download Results'. The main content area displays 'TOTAL RESULTS' as '9'. To the right, it shows '165.227.' and 'Digital Ocean Added on 2018-'. Below this, there is a 'TOP COUNTRIES' section with a world map and a list of countries: Singapore (1), Netherlands (1), India (1), United Kingdom (1), and France (1). On the right side of the map, there is a 'honeypot' button and a partial IP address '93.12.21' and '98.212.12.93.re'.

Conclusion



- With an increasing number of skilled hackers focusing on ICS, the need for more accurate ICS Honeypots is evergrowing
- A closer look at the simulation of ICS protocols and randomization of default configurations can be useful

Questions?

info@scadapot.com

YouWall.com

Disney.cn



Thanks to the following for their uninterrupted support in this research:



Ali Razmjoo Qalaei (Founder, OWASP HoneyPot, OWASP Nettacker)



Abbas Naderi Afooshteh (CEO, ZDRsearch)



©Disney