Using Cloud Honeypot Platforms for Gathering Industrial-Control-System Attack Intelligence

Alexander D. Washofsky

Neil C. Rowe

Thuy D. Nguyen

U.S. Naval Postgraduate School

Contact: ncrowe@nps.edu

Industrial control systems security

- ICSs are vulnerable attack targets because:
  - They are often critical infrastructure
  - Their software is difficult to update.
- Honeypots are essential tools for ICS security because of their specialized traffic, but you need a lot to get sufficient data.
- Cloud services are a good place to run a large set of honeypots, but:
  - ICS systems have been slow to use cloud services, so honeypots there are suspicious.
  - It's considerable work to install each honeypot separately and make them different from one another.

The research questions

- Does using electric-grid honeypots in the cloud significantly affect their traffic rates or types?
- Does use of the generic honeypot platform T-Pot affect traffic to electric-grid honeypots?
- Does location of the cloud server affect traffic to electric-grid honeypots?

- Our results showed the answers to these questions were “no” to a level of statistical significance.

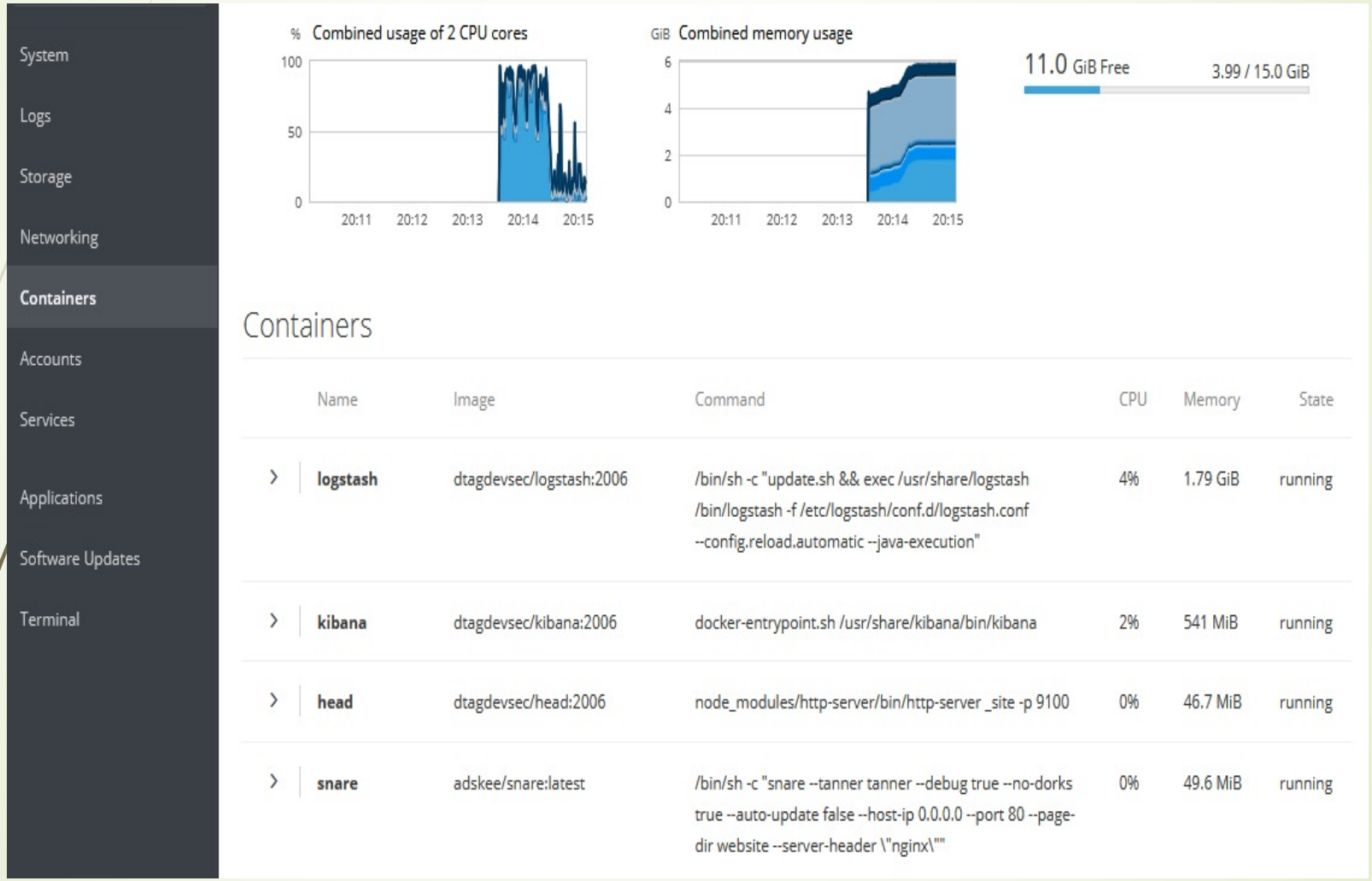
Our previous honeypot work

- One project ran the low-interaction honeypot Conpot for four months within a virtual machine on a local Internet-facing server. It serviced HTTP and several ICS protocols including EtherNet/IP, MODBUS, S7Comm, SNMP, BACnet, and IPMI.
- Another project used T-Pot to service HTTP and SSH.
- Another project connected Conpot to GridPot to emulate an electrical-distribution system.
- GridPot replaced Conpot's low-interaction handling of the IEC 60870-5-104 protocol ("IEC 104" for short) with a more interactive version that communicated to a power-grid simulator, GridLAB-D.
  - The interface was important since interfaces are the top target of attacks on ICS systems.

Our methods

- We used DigitalOcean, a cloud service unaffiliated with our school.
  - Our honeypots ran on "infrastructure as a service" virtual platforms.
- T-Pot is a honeypot deployment platform supporting several kinds of honeypots plus analysis tools.
  - We installed T-Pot on multiple DigitalOcean sites to see if it affected traffic.
  - It uses Docker virtual machines.
  - It includes Snare and Tanner for handling HTTP connections, but only rudimentary support for an ICS protocol, IEC 104 via Conpot.
  - We configured Conpot to call containerized GridPot, a physics-based ICS simulation.
  - We configured GridPot to connect to GridLab-D, an electric-grid simulation, with the IEC 104 protocol.

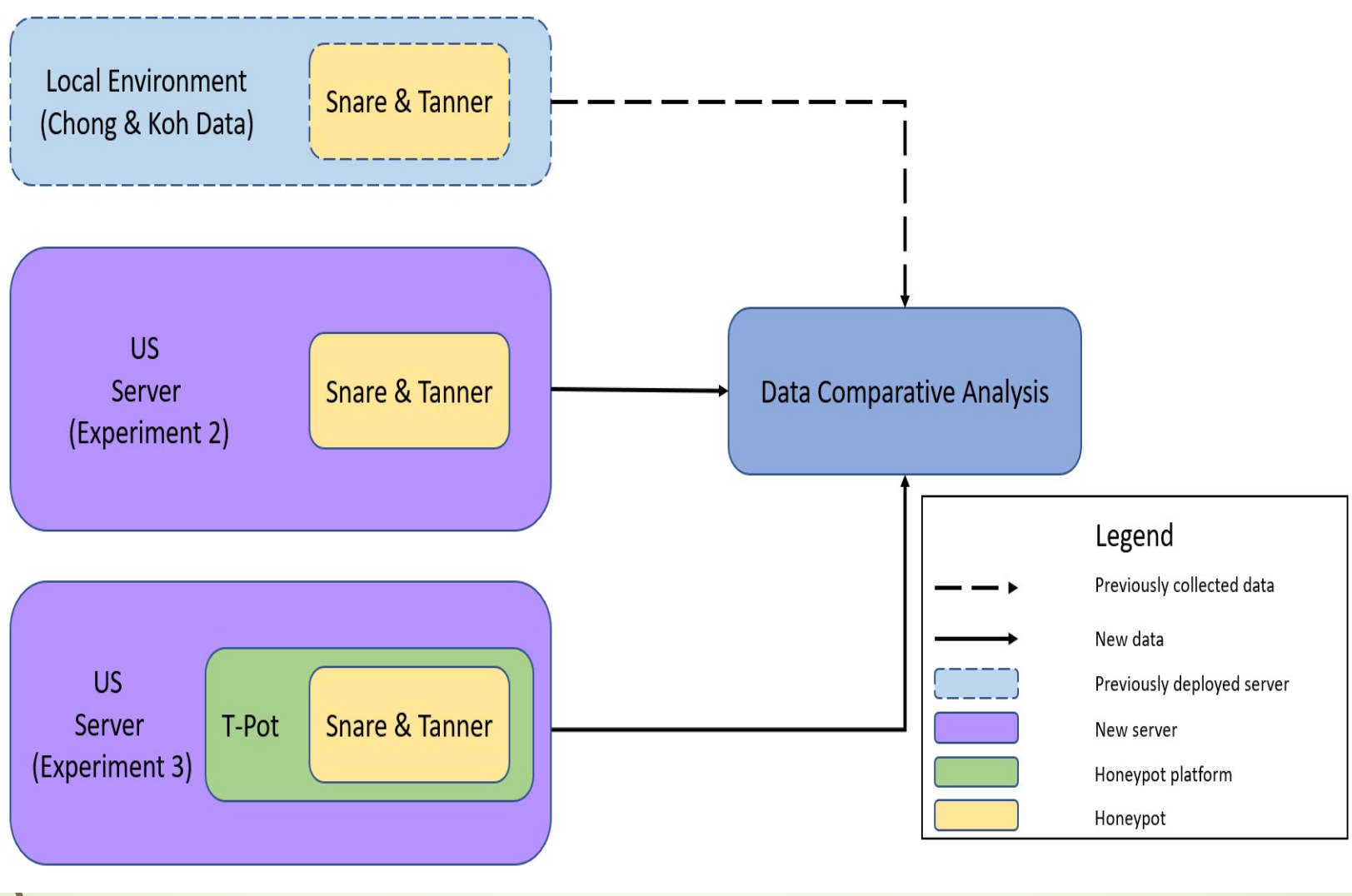
Interface to the running Docker containers with Cockpit



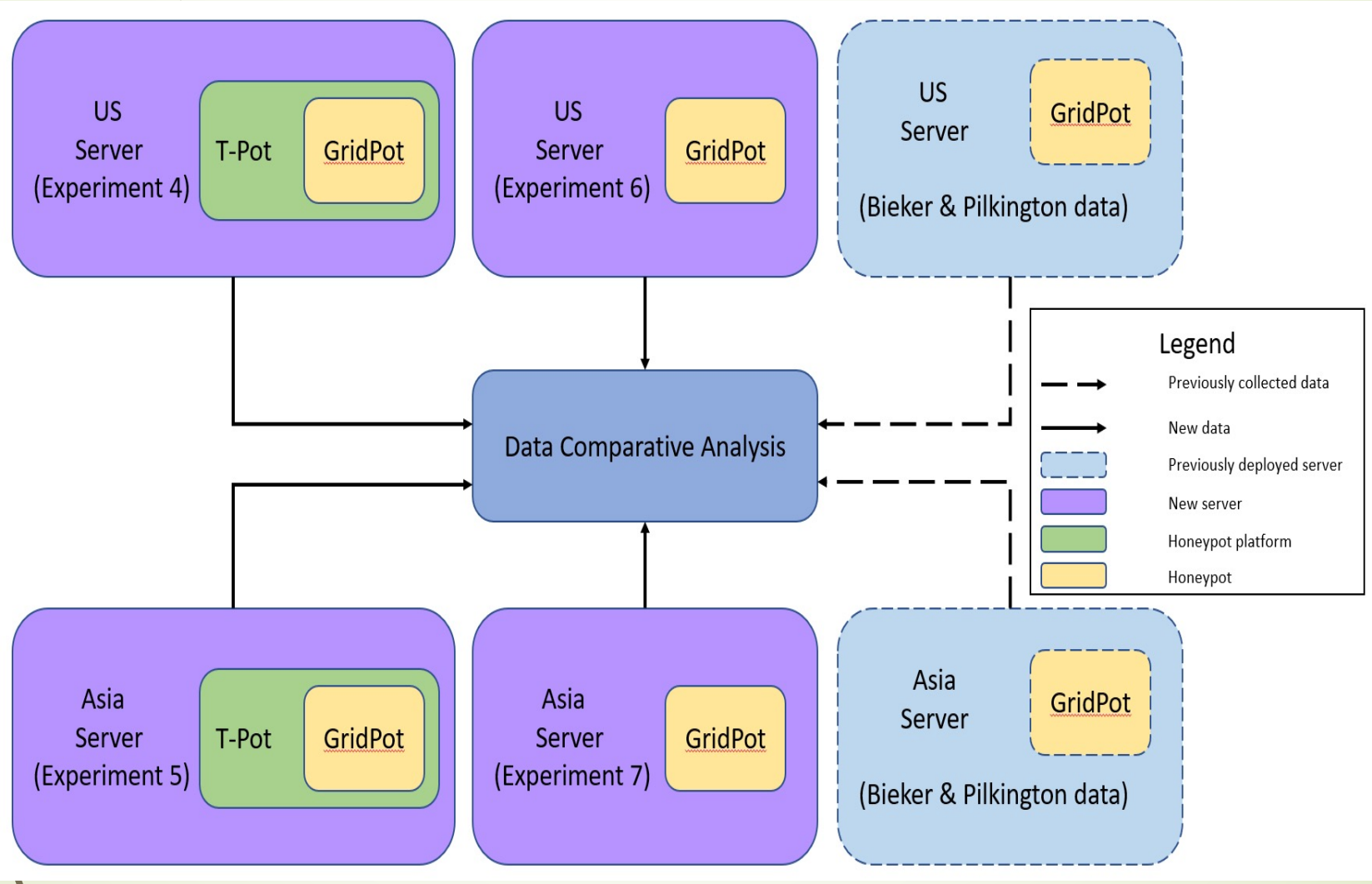
Experiments

- We did seven experiments with three deployment methods using the DigitalOcean cloud service.
- Each experiment modified the T-Pot installation script to use a custom template in a custom Docker configuration file, modified from the T-Pot default.
  - Experiment 1: Standalone Conpot honeypot inside T-Pot (compared to a version outside)
  - Experiment 2: Snare and Tanner Web honeypot outside T-Pot
  - Experiment 3: Snare and Tanner Web honeypot inside T-Pot
  - Experiment 4: Conpot and Gridpot inside T-Pot in US
  - Experiment 5: Conpot and Gridpot inside T-Pot in Asia
  - Experiment 6: Conpot and Gridpot outside T-Pot in US
  - Experiment 7: Conpot and Gridpot outside T-Pot in Asia

Setup for Experiments 2 and 3



Setup for Experiments 4-7



Obstacles encountered

- The T-Pot configuration file did not forward SNMP, BACnet, and IPMI traffic to the honeypot using UDP as it should, but by TCP.
  - After we submitted a bug report, this was corrected in Issue #781 of the T-Pot project.
- Conpot logs misclassified malformed HTTP traffic on port 80 as HTTP 0.9 packets.
- Some HTTP requests recorded by TShark packet capture were absent from the Conpot log. Some mysteriously appeared only when we stopped runs.
- Several times the setup stopped working for no apparent reason. We thus regularly logged into the machines to confirm services were running.
- Timestamps for some MODBUS and FTP packets were inaccurate.

Overall traffic observed

- Most traffic captured by our experiments was HTTP Web traffic over TCP port 80.
- We also saw traffic for ICS protocols MODBUS, EtherNet/IP, S7Comm, and IEC 104. Previous work also saw BACnet and IPMI.

Protocol	HTTP	MODBUS	Ethernet/IP	S7Comm
Count in Experiment 1	5656	1079	79	651

- Since our time was limited, we focused on responding to only HTTP and IEC 104.

Statistics on HTTP packets, Exps. 2&3

	Exp. 2 without Feb. 27 outlier	Exp. 3
Total HTTP requests	14843	10619
GET	11372 (76.62%)	8976 (84.53%)
POST	3295 (22.2%)	1436 (13.52%)
HEAD	122 (0.82%)	129 (1.21%)
CONNECT	41 (0.28%)	48 (0.45%)
OPTIONS	13 (0.09%)	27 (0.25%)
PROPFIND	0 (0 %)	3 (0.03%)

Statistics on packets, Exps. 4-7

	Exp. 4	Exp. 5	Exp. 6	Exp. 7
Unique IP addresses	834	1034	305	1509
HTTP packets	5673	5830	8343	879
IEC 104 packets	140	121	121	127
Malformed IEC 104 packets	104	95	105	110

Statistics on sessions

	Exp. 4	Exp. 5	Exp. 6 (USA)	Exp. 7 (Asia)
Unique IP addresses	320	353	305	1509
Single HTTP-only sessions	192 (60%)	211 (60%)	192 (63%)	457 (30%)
Multiple HTTP-only sessions	108 (34%)	127 (36%)	92 (30%)	1030 (68%)
Single IEC 104-only session	10 (31%)	5 (1.5%)	7 (2%)	6 (0.4%)
Multiple IEC 104-only sessions	7 (2%)	6 (1.6%)	11 (4%)	14 (0.9%)
HTTP and IEC 104 sessions	3 (0.9%)	4 (1.1%)	2 (.6%)	2 (0.1%)

Country distribution was unsurprising

	Exp. 4	Exp. 5	Exp. 4 (7 days)	Exp. 5 (7 days)	Exp. 6	Exp. 7
1	US (31.4%)	US (30.4%)	US (31.6%)	US (29.5%)	US (30.3%)	US (35%)
2	China (11.3%)	China (12.3%)	China (12.5%)	China (17.8%)	China (15.8%)	China (6.4%)
3	Germany (5.8%)	Germany (4.5%)	Germany (6.6%)	Germany (6.2%)	France (6.6%)	Russia (5.6%)
4	India (4.3%)	U.K. (4.3%)	Netherlands (5.3%)	Netherlands (5.4%)	Germany (5.9%)	Netherlands (5.2%)
5	Russia (4.1%)	Russia (4.0%)	India (4.4%)	UK (3.7%)	Russia (4.6%)	Germany (4.9%)
6	Netherlands (4.1%)	India (3.9%)	France (4.4%)	France (3.4%)	Netherlands (4.6%)	Singapore (4.4%)
7	U.K. (3.6%)	Netherlands (3.7%)	Russia (3.8%)	Russia (3.4%)	India (4.3%)	Canada (2.9%)
8	France (3.1%)	France (3.3%)	UK (3.8%)	Singapore (2.8%)	Brazil (3.0%)	France (2.7%)
9	Brazil (2.9%)	Singapore (2.8%)	Singapore (2.8%)	Brazil (2.3%)	U.K. (3.0%)	India (2.3%)
10	Other (26.6%)	Other (25.4%)	Other (19.7%)	Other (20.1%)	Other (18%)	Other (26.9%)

Web pages requested

	Exp. 4	Exp. 5	Exp. 6	Exp. 7
HTTP data	31 days	31 days	7 days	7 days
/	948 (17.9%)	1028 (19.1%)	262 (32.5%)	1880 (23.3%)
/_ignition/execution-solution	46 (0.9%)	77 (1.4%)	14 (1.7%)	16 (0.2%
/manager/html	15 (0.3%)	27 (0.5%)	3 (0.4%)	5 (0.1%)
/config/getuser?index=0	31 (0.6%)	22 (0.4%)	6 (0.7%)	6 (0.1%)
/login	11 (0.2%)	21 (0.4%)	2 (0.2%)	4 (0.1%)
/Jenkins/login	8 (0.2%)	18 (0.3%)	1 (0.1%)	4 (0.1%)
Index.html	389 (7.3%)	432 (8.0%)	102 (12.6%)	1781 (22.0%)
Category: PHP	2833 (53.4%)	2474 (46%)	110 (13.6%)	1100 (13.6%)
Category: SQL	5 (0.1%)	50 (0.9%)	1 (0.1%)	20 (0.3%)
Category: Crawler	213 (4.0%)	295 (5.5%)	67 (8.3%)	268 (3.3%)
Category: .xml	177 (3.3%)	113 (2.1%)	31 (3.8%)	18 (0.22%)
Category: Shell commands	58 (1.1%)	103 (1.9%)	8 (0.99%)	36 (0.45%)
Category: JSON	71 (1.3%)	80 (3.2%)	22 (2.7%)	23 (2.0%)
Category: Top-level folders	162 (3.0%)	224 (4.2%)	100 (12.4%)	55 (0.7%)
Category: Files	77 (1.5%)	124 (2.3%)	19 (2.4%)	49 (0.6%)
Category: JavaScript	81 (1.5%)	38 (0.7%)	0	9 (0.1%)
Category: Other .env	13 (0.2%)	82 (1.5%)	6 (0.7%)	0
Other	172 (3.2%)	162 (3.0%)	54 (6.7%)	252 (1.27%)
DNS redirect	0	0	0	2615 (32.35%)
Total	5305	5370	807	8081

Example IEC 104 trying to be HTTP


```
0000  ee 7d 86 b8 54 ed fe 00  00 00 01 01 08 00 45 08   ·}··T··· ·····E·
0010  01 03 50 e5 40 00 2c 06  3e 42 b9 b4 8f 94 80 c7   ··P·@·,· >B·····
0020  f4 b5 95 46 09 64 10 f2  c9 0a af 5c b2 40 80 18   ···F·d·· ···\·@··
0030  00 e5 db 90 00 00 01 01  08 0a 14 e0 cc 5e fd a8   ·····••• ·····^··
0040  f8 68 47 45 54 20 2f 20  48 54 54 50 2f 31 2e 31   ·hGET /  HTTP/1.1
0050  0d 0a 48 6f 73 74 3a 20  31 32 38 2e 31 39 39 2e   ··Host:  128.199.
0060  32 34 34 2e 31 38 31 3a  32 34 30 34 0d 0a 55 73   244.181: 2404··Us
0070  65 72 2d 41 67 65 6e 74  3a 20 4d 6f 7a 69 6c 6c   er-Agent : Mozill
0080  61 2f 35 2e 30 20 28 57  69 6e 64 6f 77 73 20 4e   a/5.0 (W indows N
0090  54 20 31 30 2e 30 3b 20  57 4f 57 36 34 29 20 41   T 10.0;  WOW64) A
00a0  70 70 6c 65 57 65 62 4b  69 74 2f 35 33 37 2e 33   ppleWebK it/537.3
00b0  36 20 28 4b 48 54 4d 4c  2c 20 6c 69 6b 65 20 47   6 (KHTML , like G
00c0  65 63 6b 6f 29 20 43 68  72 6f 6d 65 2f 36 36 2e   ecko) Ch rome/66.
00d0  30 2e 33 33 35 39 2e 31  31 37 20 53 61 66 61 72   0.3359.1 17 Safar
00e0  69 2f 35 33 37 2e 33 36  20 0d 0a 41 63 63 65 70   i/537.36  ··Accep
00f0  74 3a 20 2a 2f 2a 0d 0a  41 63 63 65 70 74 2d 45   t: */*·· Accept-E
0100  6e 63 6f 64 69 6e 67 3a  20 67 7a 69 70 0d 0a 0d   ncoding:  gzip···
0110  0a                                                 ·
```

Example coordinated IEC 104 activity

End of IP address	Exp. 4	Exp. 5	Exp. 6	Exp. 7
.49.96				1 U-Format
.49.97	2 U-Format, 1 I-Format			
.49.98		2 U-Format, 1 I-Format		
.49.99		3 U-Format, 1 I-Format	1 U-Format	
.49.100	2 U-Format, 1 I-Format			
.49.101	2 U-Format, 1 I-Format			
.49.102			1 U-Format	
.49.103		2 U-Format, 1 I-Format		1 U-Format
.49.104				1 U-Format
.49.109	2 U-Format, 1 I-Format		1 U-Format	
.49.110			1 U-Format	

Measuring traffic similarity

- We used unweighted cosine similarity to compare traffic between experiments using a vector of four counts:
  - HTTP GET commands, HTTP POST commands, other HTTP methods, and IEC 104 ICS traffic.
- Cosine similarity compared the proportions of traffic although volume varied widely from day to day.
- Our previous work used the ratios of cosine similarity between distributions, which has an F-distribution, but these can be inconsistent.
- However, one minus the cosine similarity looks like one side of a normal distribution centered on zero, useful for one-tail tests.
- Cosine similarity is analogous to a standard deviation between distributions.
- So, we used a modified two-sample T-test that computes significance as $\frac{2(1-s_{ij})}{e_{si}+e_{sj}}$ where e is the standard error of cosine similarities between successive weeks in experiment i.
- We used standard error rather than standard deviation because of the bursty nature of our data.

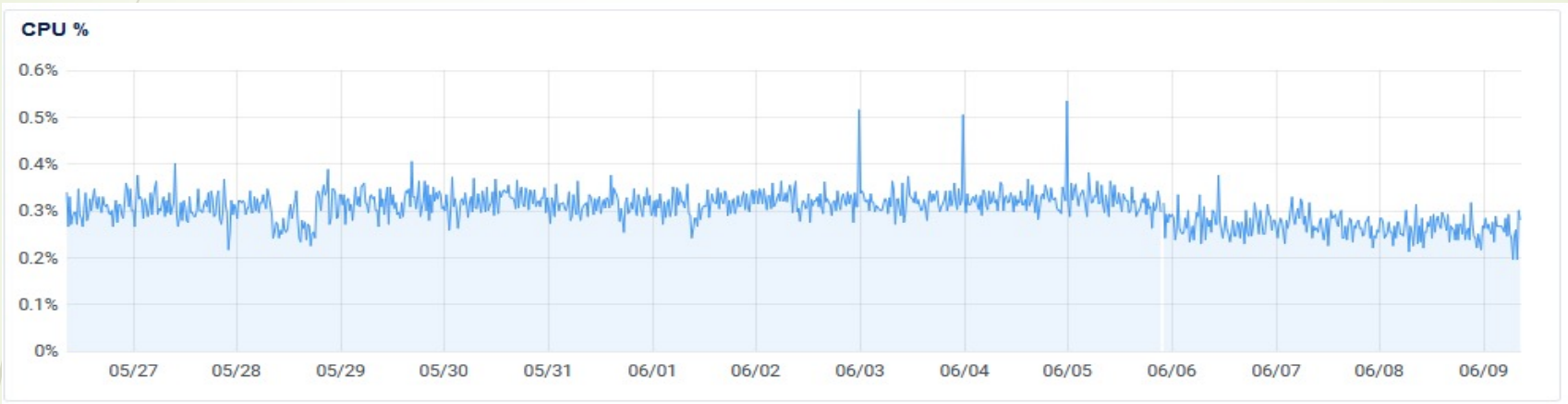
Comparing 4-count traffic vectors between experiments

	Do	Bp1	Bp2	Bp3	Wa1	Wa2	Wa3	Wa4	Wa5	Wa6	Wa7
Do	1 / .00	.809 / .25	.955 / .06	.712 / .32	.759 / .31	.836 / .21	.763 / .31	.699 / .37	.953 / .07	.737 / .35	.965 / .05
Bp1		1 /.00	.939 / .08	.988 / .01	.994 / .01	.998 / .00	.997 / .00	.985 / .02	.947 / .09	.985 / .02	.767 / .35
Bp2			1 /.00	.876 / .14	.913 / .16	.950 / .07	.910 / .23	.872 / .16	.999 / .00	.899 / .14	.938 / .09
Bp3				1 /.00	.994 / .01	.980 / .02	.997 / .00	.998 / .00	.888 / .15	.983 / .02	.667 / .42
Wa1					1/.00	.988 / .03	.997 / .00	.995 / .01	.920 / .13	.986 / .02	.733 / .40
Wa2						1 /.00	.992 / .01	.974 / .03	.960 / .06	.971 / .04	.786 / .32
Wa3							1.0 /0	.994 / .01	.921 / .13	.984 / .02	.718 / .42
Wa4								1 /.00	.882 / .18	.889 / .01	.667 / .46
Wa5									1 /.00	.902 / .16	.924 / .14
Wa6										1 /.00	.722 / .42
Wa7											1 /.00

Numbers are cosine similarity and degree of significance.

T-Pot requires more processing

Experiment 2 (standalone honeypot, top) used 20-25 times less processing power, six times less memory, and four times less disk space than Experiment 3 (a T-Pot installation, bottom).



Conclusions

- Overall, differences were not significant between:
  - Local honeypots and honeypots deployed in the cloud
  - Honeypot deployed within T-Pot and without
  - Honeypots deployed in the U.S. versus in Asia
- Most interactions with our honeypots appeared to be cyberattacks, not scanning, for both HTTP and IEC 104 traffic.
  - The many packet errors do not suggest scanning.
  - HTTP interactions were predominantly privilege escalation attempts.
- These results support using honeypots with greater flexibility to enable better deception of possible attackers.
- Our T-Pot deployment was not fingerprinted by Shodan, a well-known network scanning tool.
- Current work is running the honeypots longer and in other more-hardened frameworks, generating realistic attack traffic, and doing machine learning on the data.