Honeypots and Machine Learning: *an effective tag team*

David Read
Senior Lecturer, Department of Computer Science

Celebration WEEKEND

Agenda

1. Honeypots
2. Machine Learning
3. Finding Attack Patterns

Why Worry About Data Security?

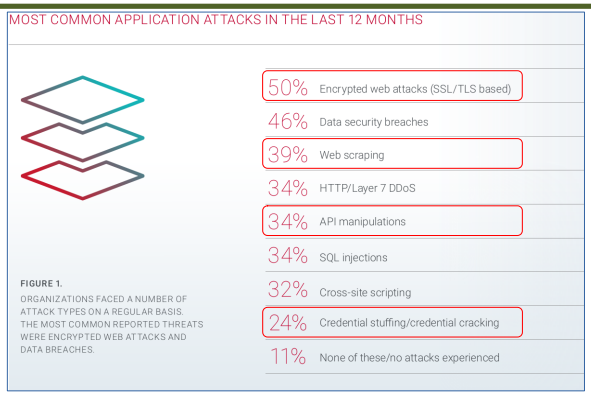
More than 25% experienced attacks on a daily basis

~50% indicated that some of their customers asked for compensation or their own reputations suffered because of application/web server attacks.

Respondents said that data security breaches were the most difficult type of application attack to detect and mitigate.

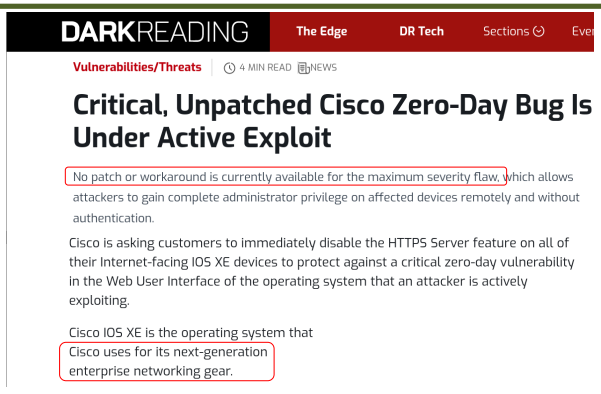
Source: The State of Web Application Security, radware, 2018: a survey of 300 IT leaders at large companies across APAC, AMER, and EMEA

Common Attacks



Source: The State of Web Application Security, Radware, 2018: a survey of 300 IT leaders at large companies across APAC, AMER, and EMEA

Everyday...



Source: https://www.darkreading.com/vulnerabilities-threats/critical-unpatched-cisco-zero-day-bug-active-exploit (10/16/2023)

Computer Connections

Most networks use the Ethernet standard
  First implemented late 1960s early 1970s

Each computer has an address and ~65000 ports (doorways) through which they can communicate (information in and out)

Most ports are generally closed and boarded-up

Only one operation may use a port at a time

For open ports, security? Not so much
  You ring the doorbell, I'll let you in
  Fake id, that's fine

Good Connection, Bad Connection?

Computers are expected to make and receive connections - that's the point of a network

Google, Amazon, Instagram… they receive 1000s of connection requests each second during peak times

Ethernet defines how a connection is established, whether your intent is legitimate or diabolical, you must connect in the same way

Very difficult to identify a bad actor at this point in the process… *if I knock on your door you have to let me in before you'll know my intent*

Voila, the Honeypot

How can this conundrum be resolved?

We want to know as soon as a connection is attempted if the intention is to do harm

What if the act of connecting was itself an attack? In other words, what if no computer should ever try to connect to our computer?

A honeypot!

A computer with no legitimate business purpose, other than detecting attackers looking for computers to hack into.

Sensor Readings, Mr. Spock

Since every access to the honeypot is an attack we want to record the attacker's actions and learn about their approaches to breaking in.

However, the attacker won't stay engaged with the honeypot if it doesn't look like a useful computer with interesting applications and valuable data.

Sensors are programs which record the attacker's actions. Many of them mimic common computer services but don't return real business or personal information.

Sensor(y) Overload

Sensors make setting up a honeypot server fairly simple:

  Conpot, cowrie, p0f, elasticpot, dionaea, suricata, medpot…

Can get carried away and deploy lots of them on a honeypot system, but…

  Deploy too many and the honeypot becomes obvious because most computers are setup to run only a few business services.

Honeypot Placement: External/DMZ

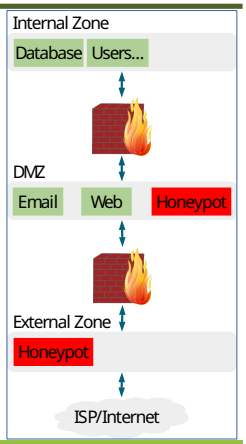
Focus on detecting external threat actors

Will see lots of "noise"

Port scans, script kiddie activity

Good for watching the front door

Need to be vigilant that it doesn't become a base for bad actors on the Internet



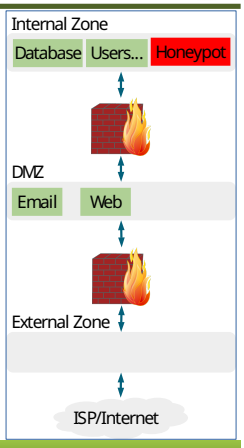
Honeypot Placement: Internal

Focus on detecting internal threat actors (including breached systems)

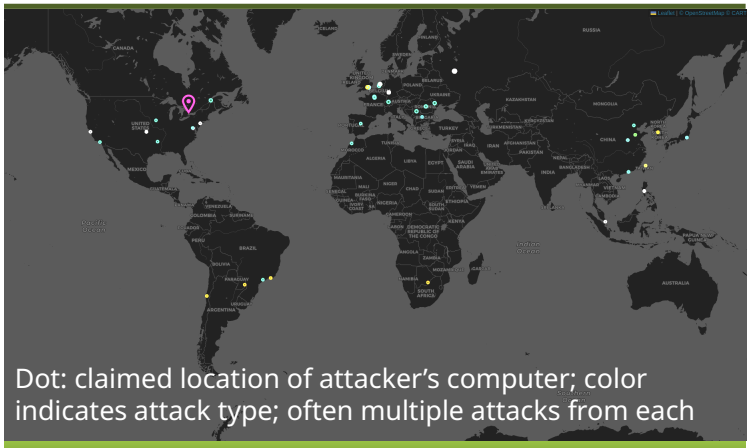
Should be very quiet

Activity is very likely a rogue insider or compromised system

Need to be vigilant that it doesn't become a base for bad actors on your network

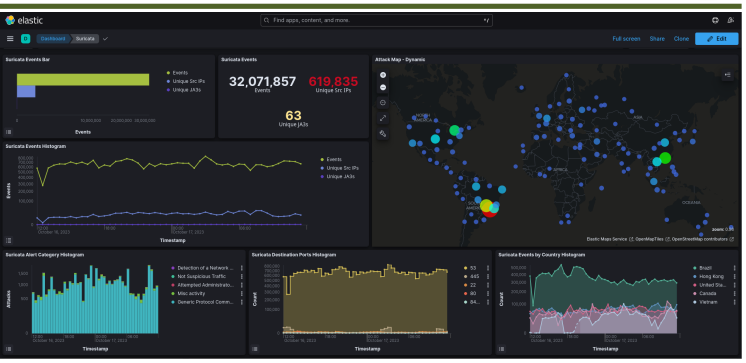


Overview from Honeypot



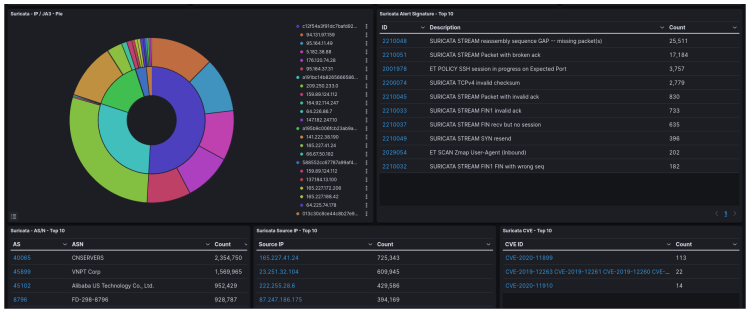
Dot: claimed location of attacker's computer; color indicates attack type; often multiple attacks from each

Summary from Suricata Sensor

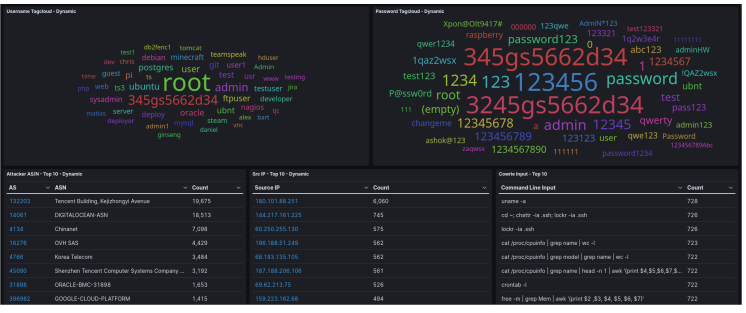


This is showing data for a 24 hour period.
32 million attacks, from 600,000 IP addresses

Well-known Attacks Report from Suricata



IDs and Passwords from Cowrie Sensor



Cowrie mimics ssh and telnet, common protocols for logging into a computer. The sensor records the attempted credentials and commands.

Honeypots In The News

Kaspersky honeypots find 105 million attacks on IoT devices in first half of 2019

The number of attacks on IoT devices in 2019 is nine times greater than the number found in the first half of 2018.

After deploying more than 50 honeypots worldwide, Kaspersky detected 105 million attacks on Internet of Things (IoT) devices from 276,000 unique IP addresses, within only the first six months of 2019. The number of attacks in 2019 is nine times greater than the number found in the first half of 2018, which totaled 12 million attacks.

Kaspersky's IoT: A Malware Story report, released on Tuesday, used honeypot data to determine the number of cyberattacks conducted in the time frame, which type of attacks were used, and where these attacks took place. As organizations purchase more connected smart devices, attackers find more threat vectors to target, the report said.

What are honeypots?

A tool used by many security experts, honeypots are decoys used to mimic typical targets of attack and subsequently attract cyberattackers, as Jack Wallen reported.

Kaspersky incorporated three common types of honeypots: Low-interaction, high interaction, and medium interaction. The first simulates services such as Telnet, SSH, and web servers; the second mimics real devices, and the third is a mixture of the two.

To avoid being discovered quickly by cybercriminals, Kaspersky's honeybots cycled through IP addresses often. Some honeypots kept the same

Source: https://www.techrepublic.com/article/kaspersky-honeypots-find-105-million-attacks-on-iot-devices-in-first-half-of-2019/

Accessed: 2019-10-18

Honeypot Summary

Each interaction is an attack

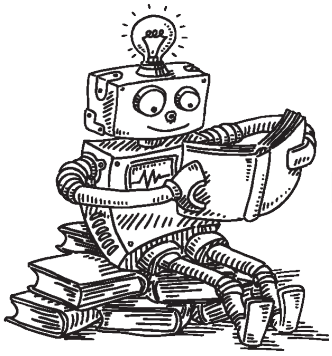
Sensors allow us to collect details about the attacker's actions

Location of the honeypot on the network affects the attacker threat level we will detect

May generate lots of data very quickly

Data will contain clues about attacker's actions and tools

Want to use this information to protect real systems – How?



Machine Learning

Two Major Types of Machine Learning

Supervised: take training data that contains "answers" and have the computer figure out how to predict those answers – *build a model*

Whitebox: we can understand the model

Black box: the model is a mystery to us

Unsupervised: take a set of data without predetermined relationships and have the computer identify relationships or groupings

Data Cleanup

Always a large task in any machine learning project

Key challenge with honeypot data is that sensors are created by different groups leading to a lack of consistency

- Data fields with the same information referred to by different names

Need to unify terminology, data structure, time periods, and aggregation

Sample of Raw Sensor Data Differences

p0f data sample (connection attempts sensor)

{"payload" : {"client_ip": "XXX.171.255.51", "dist" : "23", "server_port" : 81, "timestamp" :"2018/02/12 22:12:51", "client_port" : 19468, "raw_sig" : "4:232+23:0:0:14600,0:::0", "params" : "none", "server_ip" : "XXX.227.155.118", "mod" : "syn", "os" : "???", "subject" : "cli" }}

cowrie data sample (SSH honeypot sensor)

{"payload" : {"peerIP": "XXX.224.48.246", "commands" : [ ], "loggedin" : null, "protocol" : "ssh", "urls" : [ ], "ttylog" : null, "hostPort" : 22, "peerPort": 44584, "session" : "0effd545c86c", "startTime": "2018-02-12T22:42:17.373041Z", "hostIP" : "XXX.227.155.118", "credentials" : [ ], "hashes" : [ ], "endTime": "2018-02-12T22:42:17.564252Z", "version" : null, "unknownCommands" : [ ] }}

Sample of Cleaned Sensor Data

p0f data sample (connection attempts sensor)

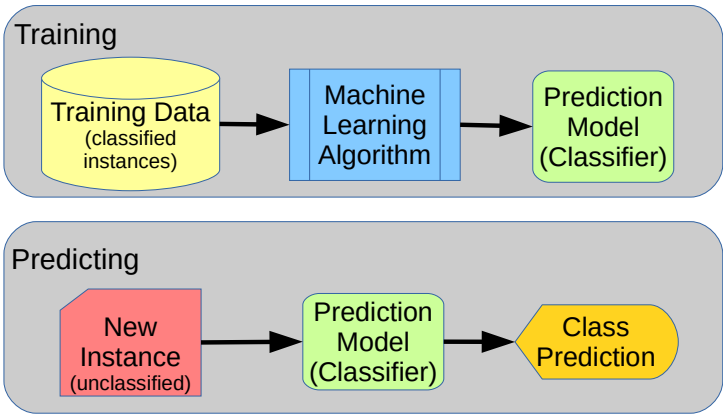
{ "_id" : ObjectId("5a8210ddb393ea05ab7de799"), "timestamp": ISODate("2018-02-12T22:10:37.167Z"), "normalized" : true, "channel" : "p0f.events", "clientIP" : "197.251.253.73", "serverIP": "165.227.155.118", "clientPort" : 52690, "serverPort" : 22, "hour" : 22, "day" : 1, "hour8" : 3, "hour8EastUS" : 2, "hour8CN" : 4 }

cowrie data sample (SSH honeypot sensor)

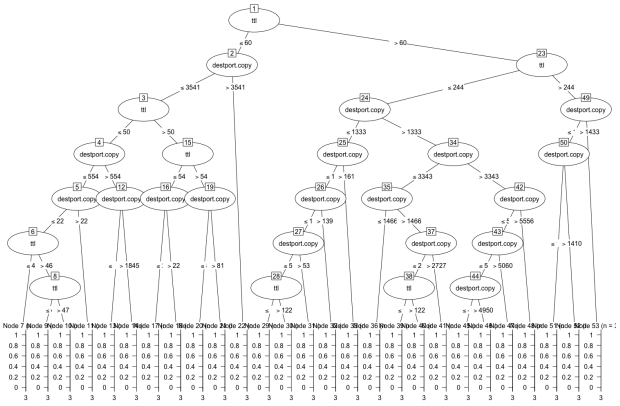
{ "_id" : ObjectId("5a8210e0b393ea05ab7de79c"), "timestamp": ISODate("2018-02-12T22:10:40.807Z"), "normalized" : true, "channel" : "cowrie.sessions", "clientIP": "197.251.253.73", "serverIP" : "165.227.155.118", "clientPort" : 52690, "serverPort" : 22, "hour" : 22, "day" : 1, "hour8" : 3, "hour8EastUS" : 2, "hour8CN" : 4 }

Finding Attack Patterns

The Supervised Training Process



Example Supervised Model (D. Tree)

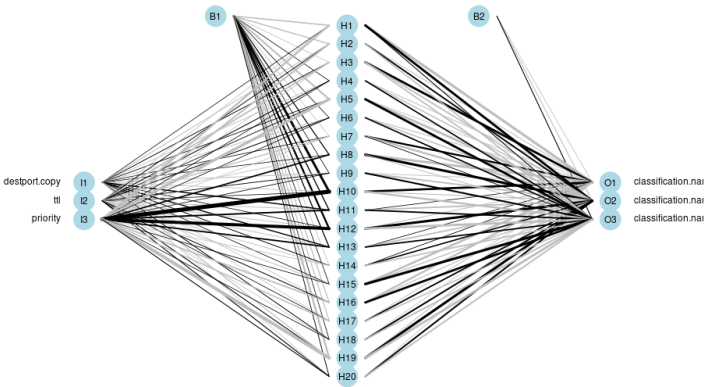


Example Supervised Model (D. Tree)

```
Correctly Classified Instances        1241               98.0253 %
Incorrectly Classified Instances        25                1.9747 %
Kappa statistic                          0.9704
Mean absolute error                      0.0242
Root mean squared error                  0.11
Relative absolute error                  5.4466 %
Root relative squared error             23.3379 %
Total Number of Instances             1266

=== Confusion Matrix ===

   a   b   c   <-- classified as
 418   0   4 |   a = 3
   0 415   7 |   b = 4
   7   7 408 |   c = 30
```

```
Confusion Matrix and Statistics

          Reference
Prediction   3   4  30
         3 275   0   6
         4   0 273   8
         30   6  10 265

Overall Statistics

               Accuracy : 0.9644
                 95% CI : (0.9496, 0.9759)
    No Information Rate : 0.3357
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9466
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 3 Class: 4 Class: 30
Sensitivity            0.9786   0.9647    0.9498
Specificity            0.9893   0.9857    0.9716
Pos Pred Value         0.9786   0.9715    0.9431
Neg Pred Value         0.9893   0.9822    0.9751
Prevalence             0.3333   0.3357    0.3310
Detection Rate         0.3262   0.3238    0.3144
Detection Prevalence   0.3333   0.3333    0.3333
Balanced Accuracy      0.9840   0.9752    0.9607
```


Example Supervised Model (ANN)



Example Supervised Model (ANN)

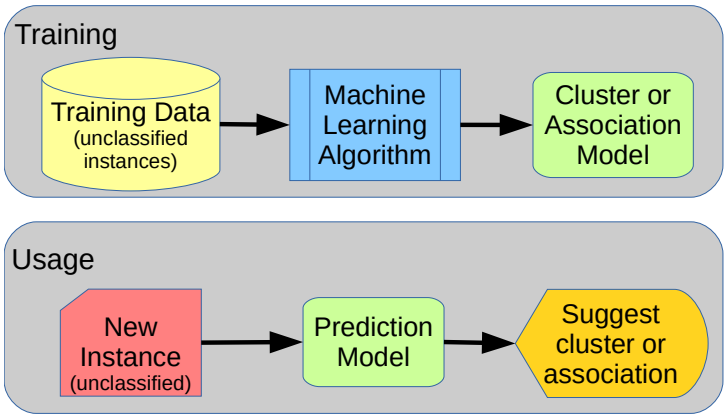
Training Data

```
          prediction
actual     A     B     C
     A   408     0    14
     B     0   412    10
     C    20     5   397
```

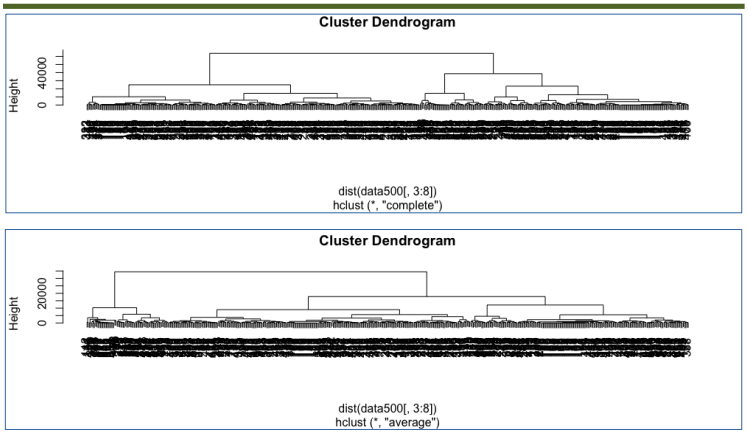
Testing Data

```
          prediction
actual     A     B     C
     A   268     0    13
     B     0   274     7
     C    13    12   256
```

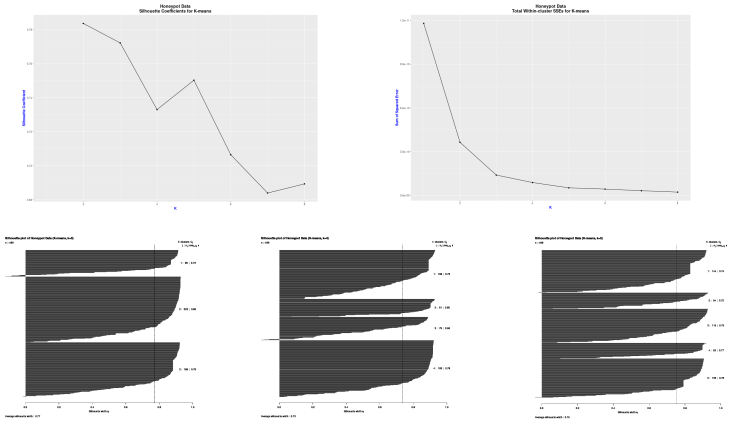
The Unsupervised Training Process



Example Unsupervised Model (H. Clust)



Example Unsupervised Model (K-means)



Prediction

Once we have a model, we run new data through it for classification (prediction)

- Interaction data from production servers

The model classifies the production interactions as nominal or suspect

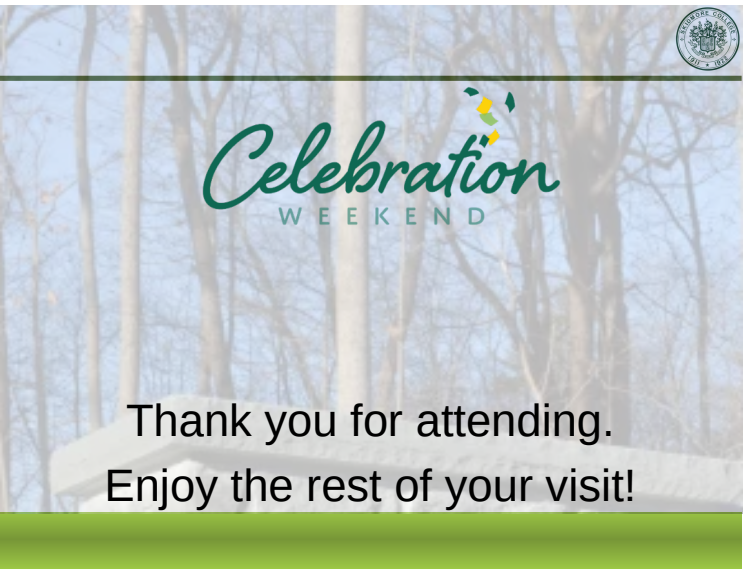
Interactions flagged as suspicious are checked to see if they represent a threat to the system

- An alarm system to warn us that an attacker may be trying to access the system

Look at a Live Honeypot

Wrap-up

- Honeypots give us a way to collect known-attacker actions
- Machine learning looks for relationships and patterns in large sets of data
- Patterns identified from honeypot interactions can be used to flag similar interactions on production systems, thereby reducing risk by protecting data, personal and corporate, as well as company reputation


Celebration
WEEKEND
Thank you for attending.
Enjoy the rest of your visit!