

CSec15233

Malicious Software Analysis

Malware Analysis Primer

Qasem Abu Al-Haija

The Goals of Malware Analysis

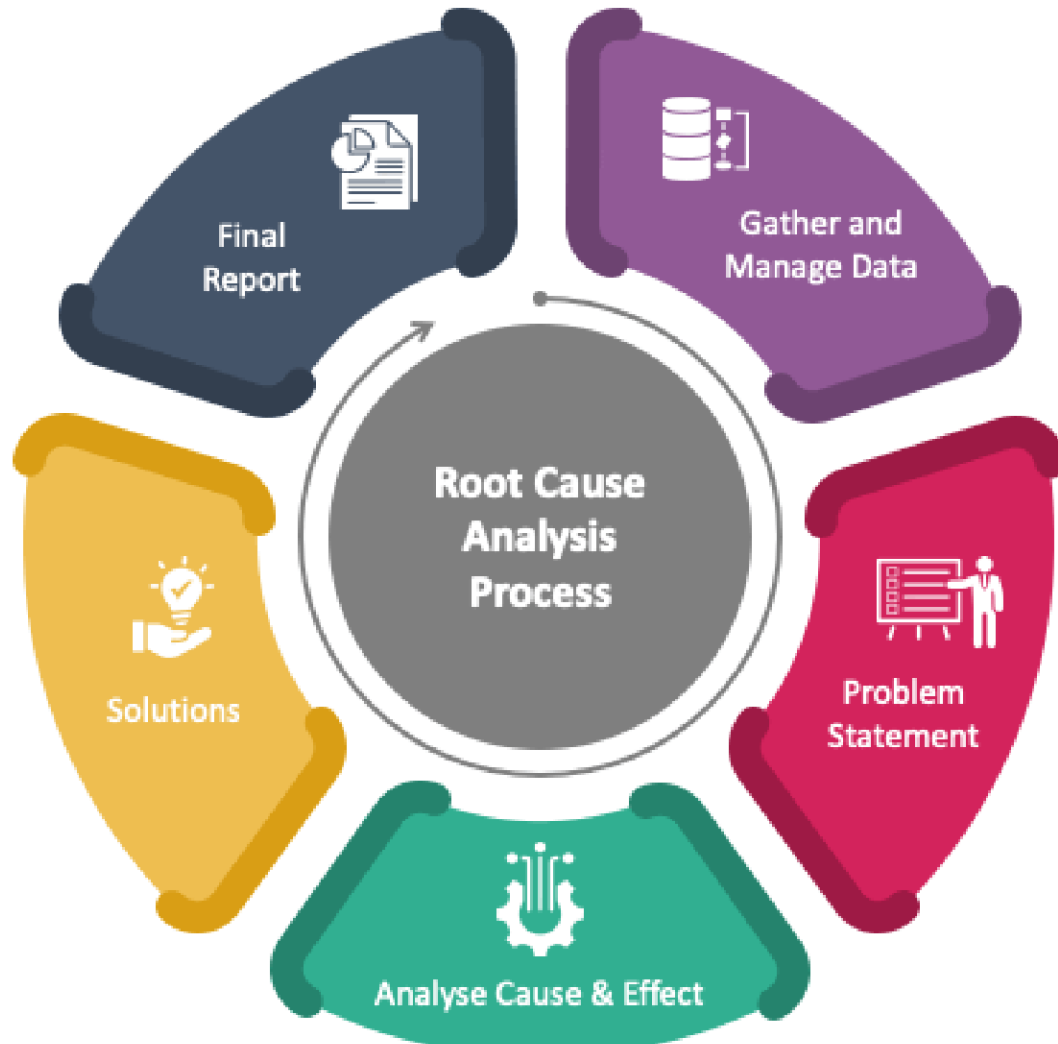
Incident Response

- Case history
 - A medical clinic with 10 offices found malware on one of their workstations
 - Hired a consultant to clean & re-image that machine
- All done—case closed?

Incident Response

- After a malware is found, you need to know
 - Did an attacker implant a rootkit or trojan on your systems?
 - Is the attacker really gone?
 - What did the attacker steal or add?
 - How did the attack get in
 - Root-cause analysis

Root-cause analysis Process




Breach clean-up cost LinkedIn nearly \$1 million, another \$2-3 million in upgrades

Summary: LinkedIn executives reveal on quarterly earnings call just what the June theft of 6.5 million passwords cost the company in forensic work and on-going security updates.



By John Fontana for Identity Matters | August 3, 2012 -- 17:10 GMT (10:10 PDT)

 Follow @johnfontana

Comments

0



Vote

1



Like

4



Tweet

51



Share

more +

LinkedIn spent nearly \$1 million investigating and unraveling the theft of 6.5 million passwords in June and plans to spend up to \$3 million more updating security on its social networking site.

Malware Analysis

- Dissecting malware to understand
 - How it works
 - How to identify it
 - How to defeat or eliminate it
- A critical part of incident response

The Goals of Malware Analysis (MA)

- **The purpose of MA is usually to provide the information needed to respond to a network/ computer intrusion.**
 - Exactly what happened
 - how to detect it on your network,
 - Ensure you've located all infected machines and files
 - How to measure and contain the damage
 - Find **signatures** for intrusion detection systems

Malware Signatures

- A specific pattern that allows recognition of malicious threats.
- Different from antivirus signatures
 - For example, Antivirus software uses a virus signature to find a virus in a computer file system.
 - Allowing for detection, quarantine, and removal of the virus.

Malware Signatures

Examples of viruses' string signature

It's like identifying a criminal by having a sample of their DNA.

<i>Virus Name</i>	<i>String Pattern (Signature)</i>
Accom.128 0	89C3 B440 8A2E 2004 8A0E 2104 BA00 05CD 21E8 D500 BF50 04CD
Die.448	B440 B9E8 0133 D2CD 2172 1126 8955 15B4 40B9 0500 BA5A 01CD
Xany.979	8B96 0906 B000 E85C FF8B D5B9 D303 E864 FFC6 8602 0401 F8C3

Malware Signatures

- Once you identify which files require full analysis:
 - it's time to develop signatures to detect malware infections on your computer or network.
 - Host-based signatures
 - Network-based signatures

Malware Signatures

- **Host-based signatures**

- Indicators used to detect malicious code on victim computers
 - Monitor computer processes
 - Identify files or registry keys on a victim's computer that indicate an infection
- Focus on what the malware did to the system, not the malware itself
 - Different from antivirus signature

Malware Signatures

- **Network signatures**

- Indicators used to detect malicious code by monitoring network traffic.

- Detect malware by analyzing network traffic

- Can be created without malware analysis.

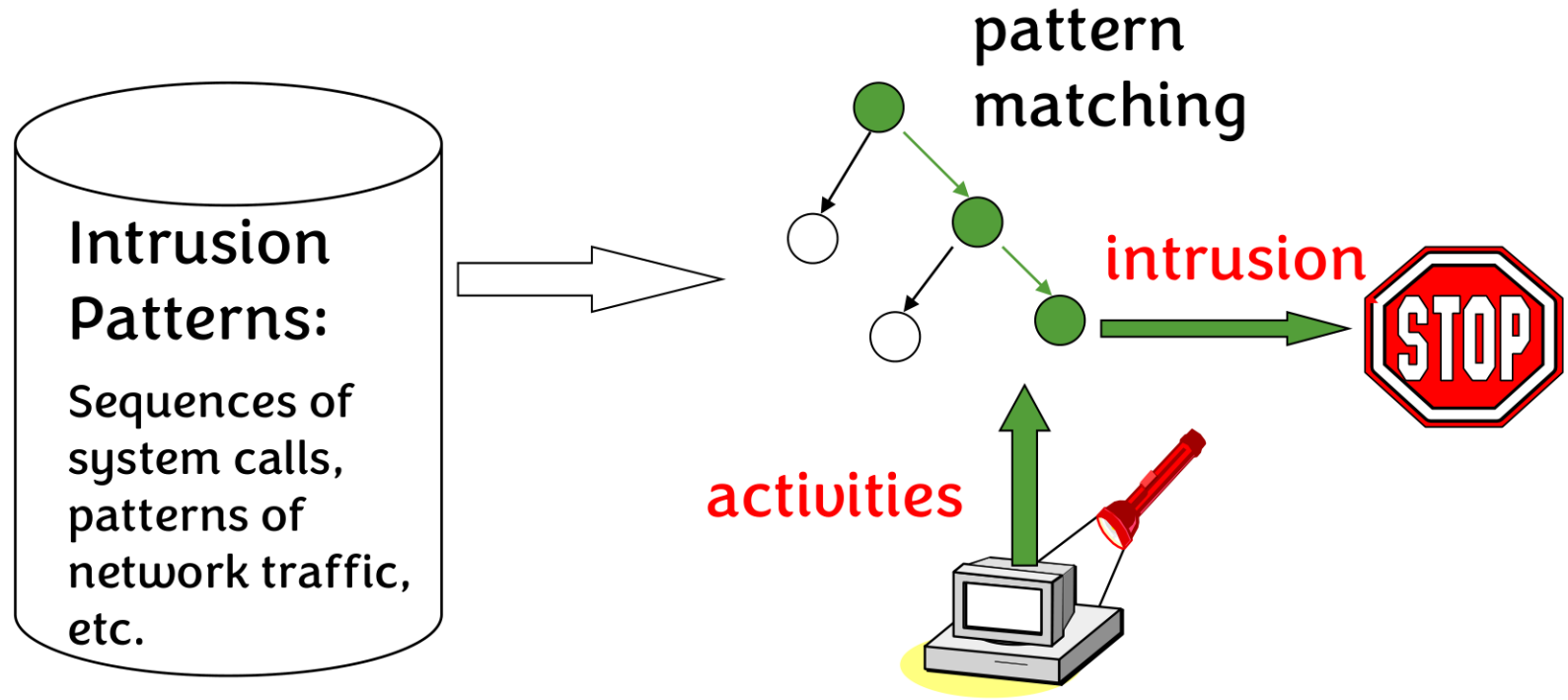
- But it's more effective when made using malware analysis.
 - Higher detection rate and fewer false positives.

Malware Signatures

- **After obtaining the signatures,**
 - the final objective is to figure out exactly how the malware works.
 - This is often the most asked question by senior management, who want a full explanation of a major intrusion.

Malware Detection using Signatures

Also called: Misuse Detection.



Example: *if* (traffic contains “x90+deZ^\r\n]{30}”) *then* “attack detected”

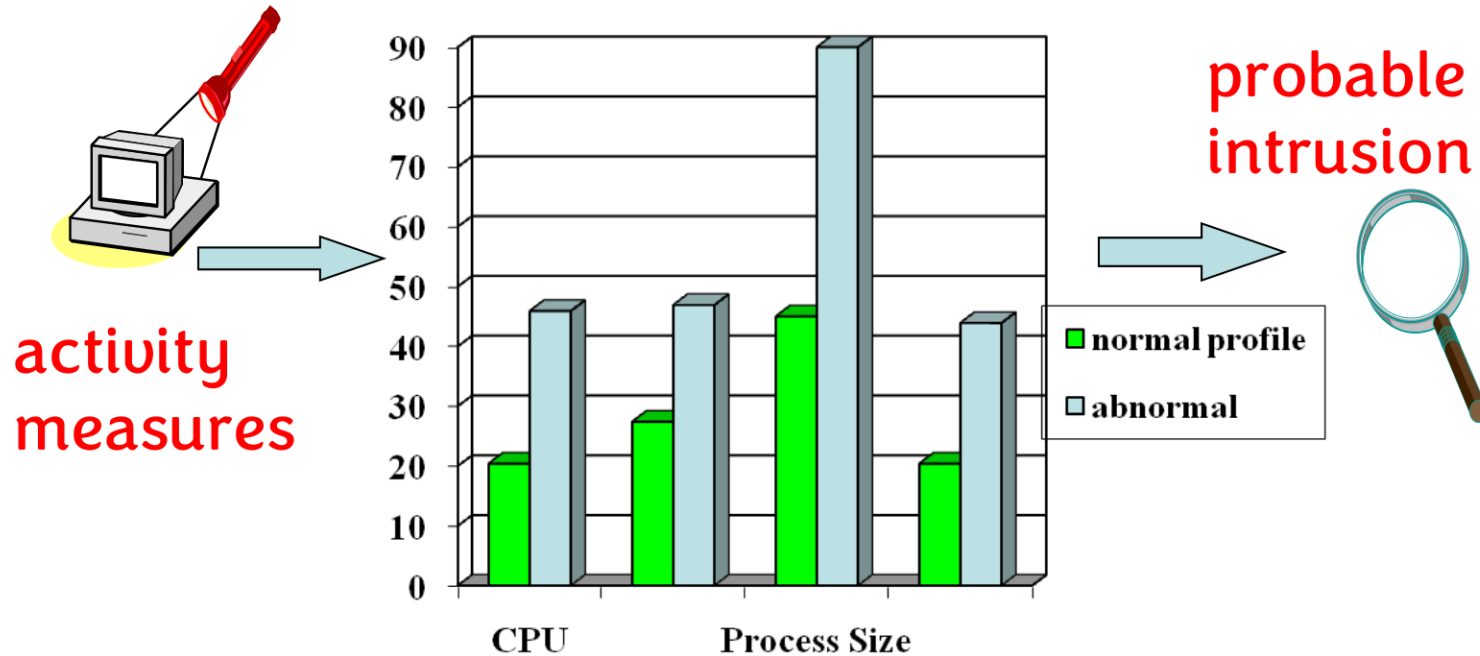
Advantage: Mostly accurate. But problems?

Can't detect new attacks

Possible Solution: **Anomaly-Based Detection**

Anomaly-based Malware Detection

Also called: statistical-based



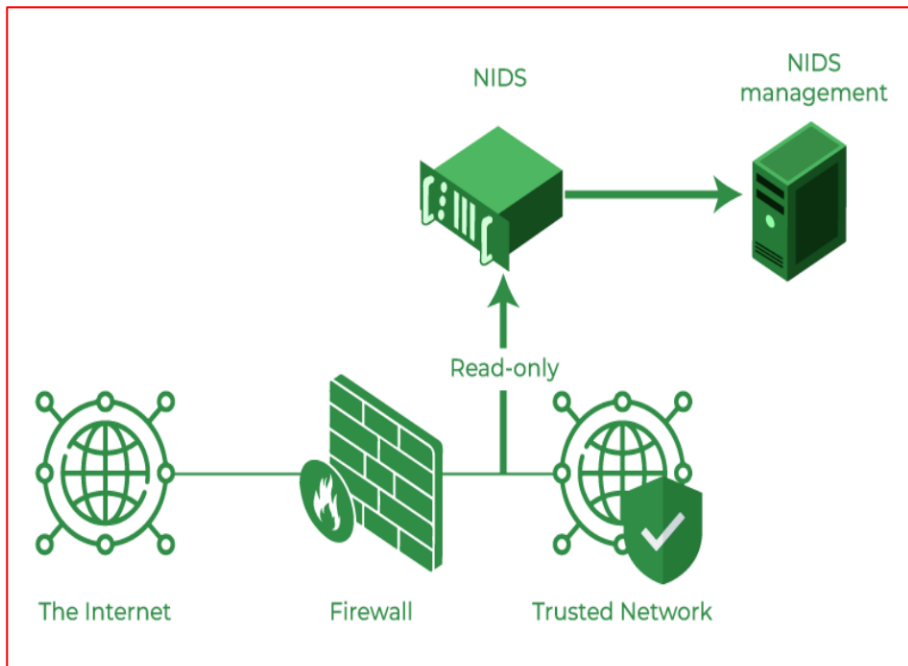
Define a **profile** describing “normal” behavior, then detects deviations. Thus, can detect potential new attacks.

- Any problem?** *Relatively high false-positive rates*
- ✓ *Anomalies can just be new normal activities.*
 - ✓ *Anomalies caused by other element faults*

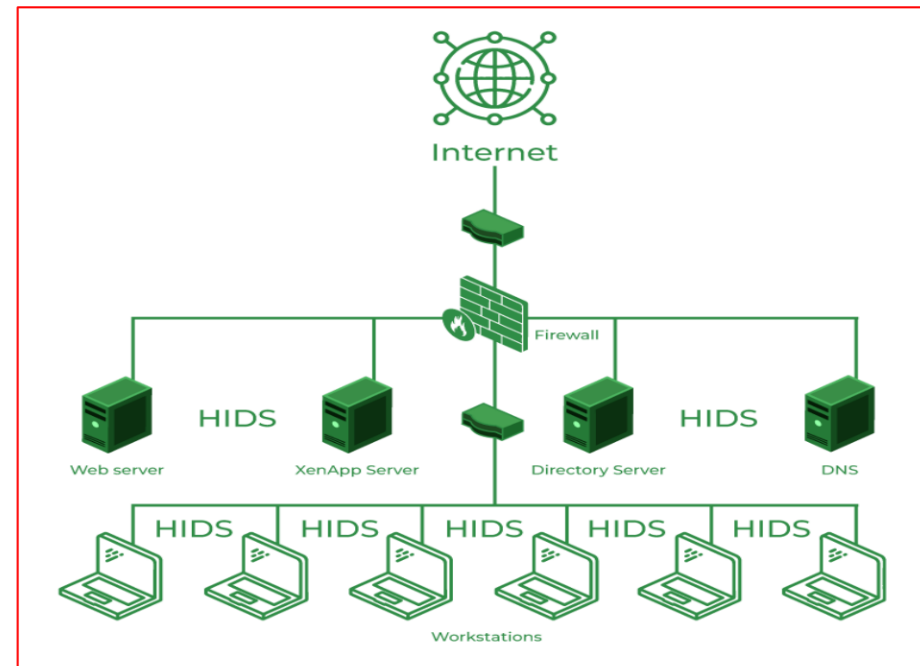
Malware Signatures

Two deployments

NIDS




HIDS



symantec.








False Positives


CBS San Francisco Your Home Buy Tickets More FOLLOW US   LOGIN

City College Of San Francisco Computer Lab Security Breached

January 13, 2012 1:56 PM

Share this  1  3  0  2 

 Share CBS Local with your friends. Add us to your Timeline. [What's this?](#)





SAN FRANCISCO (KCBS) – The personal banking data from thousands of City College of San Francisco students, faculty and staff may be at risk because of a virus that infiltrated one [computer](#) lab – perhaps years ago.

Incredibly, the breach was only discovered recently – over the Thanksgiving holiday weekend.

KCBS' Holly Quan Reports:

City College of San Francisco (CCSF)



 **Click here to play audio**

What's most disturbing isn't that the IP addresses identified as receiving transmissions belong to the Russian Mafia –

Sponsored Links
 [\\$28/Hr Data Entry Jobs At](#)

Malware Analysis Techniques

Malware Analysis Techniques

- Commonly, when performing MA, you'll have malware executable, not human-readable.
 - To make sense of it, you'll use various tools
 - Each tool reveals a small amount of information.
 - Using a variety of tools to see the full picture.
- There are two fundamental approaches to MA:
 - Static analysis and Dynamic analysis.
 - Both techniques can be basic or advanced.

Malware Analysis Techniques

Static v. Dynamic Analysis

- **Static Analysis**

- Examines malware without running it
- Tools: VirusTotal, strings, and a disassembler like IDA Pro.

- **Dynamic Analysis**

- Run the malware and monitor its effect
- Use a virtual machine and take snapshots
- Tools: RegShot, Process Monitor, Process Hacker.

Malware Analysis Techniques

Basic Static Analysis

- Examining executable files without viewing actual instructions.
- Goals
 - confirm whether a file is malicious,
 - provide information about its functionality, and
 - Sometimes provide information to produce simple network signatures
- Advantage
 - straightforward and can be quick
- Disadvantages
 - Ineffective for advanced malware and can miss important behavior
- Tools:
 - VirusTotal, strings, BinTex, HashCalc, ...

Malware Analysis Techniques

Basic Dynamic Analysis

- Running the malware and observing its behavior on the system in order
- Goals
 - remove the infection,
 - produce effective signatures, or
 - both.
- Advantage
 - can be used by people without deep programming knowledge
 - Easy but requires a safe test environment
- Disadvantages
 - won't be effective with all malware and can miss important functionality
- Tools:
 - RegShot, Process Monitor,

Malware Analysis Techniques

Advanced Static Analysis

- Reverse-engineering the malware's internals by loading the executable into a **disassembler** and looking at the program instructions.
- Goals
 - discover what the program does,
- Advantage
 - tells exactly what the program does
- Disadvantages
 - has a **steeper learning curve** than basic static analysis and
 - Complex, requires specialized knowledge of assembly code, disassembly, code constructs, and Windows OS concepts
- Tools:
 - IDA Pro

Malware Analysis Techniques

Advanced Dynamic Analysis

- Uses a **debugger** to examine the internal state of a running malicious executable. (Run code in a debugger)
- Goals
 - discover what the program does,
- Advantage
 - provide another way to extract detailed information from an executable
- Disadvantages
 - has a steeper learning curve and
 - Complex, requires specialized knowledge of assembly code, disassembly, code constructs, and Windows OS concepts
- Tools:
 - OlyDbg

General Rules for Malware Analysis

General Rules for Malware Analysis

- **Don't Get Caught in Details**
 - You don't need to understand 100% of the code
 - Focus on key features
- **Try Several Tools**
 - If one tool fails, try another
 - Don't get stuck on a hard issue; move along
- **Malware authors are constantly raising the bar.**
 - As new MA techniques are developed, malware authors respond with new techniques to thwart analysis.
 - Thus, you need to keep your self up to date.

IDS Performance Evaluation using Confusion Matrix

Confusion Matrix

- A specific table layout that allows visualization of the performance of an algorithm.
- Has four components
 - The number of True Positive (TP) samples.
 - The number of True Negative (TN) samples.
 - The number of False Positive (FP) samples.
 - The number of False Negative (FN) samples.
- Has two dimensions components
 - Actual Class (Columns)
 - Predicted Class (Rows)

Please Always consider the CONFUSION MATRIX arrangement as in the next slide.
(Don't use any other arrangement)

Confusion Matrix

		ACTUAL	
		Positive	Negative
P R E D I C T E D	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Confusion Matrix







A **True Positive** is where we predict a positive result and are **correct** in our prediction

A **False Positive** is where we predict a positive result and are **incorrect** in our prediction. This is often called a **Type 1 Error**

A **False Negative** is where we predict a negative result and are **incorrect** in our prediction. This is often called a **Type 2 Error**

A **True Negative** is where we predict a negative result and are **correct** in our prediction.

Confusion Matrix

	Actual Result	Predicted Result
	Pass	Pass
	Pass	Fail
	Fail	Pass
	Fail	Fail
	Pass	Pass
	Fail	Fail



		Actual Result	
		Pass (+)	Fail (-)
Predicted Result	Pass (+)	2	1
	Fail (-)	1	2

Our model correctly predicted the results of 4 students, from the total of 6 - giving a **Classification Accuracy** of...

$$4 / 6 = \mathbf{66.6\%}$$

Correct	4
Incorrect	2
Total	6



Accuracy

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

Actual Values

		Actual Values	
		Malware	Normal
Predicted Values	Malware	540	150
	Normal	110	200

$$ACC = \frac{540 + 200}{540 + 200 + 150 + 110} = 74\%$$

Imbalanced Testing

While **Classification Accuracy** is very intuitive, there are scenarios where it can be misleading. It is important to know when this can happen, and what metrics are more appropriate!

We can see from the Confusion Matrix that **only 2 patients out of our set of 100 actually do have the disease!**

Example:



This is what we call **Imbalanced Data!**

Imbalanced Testing

Think about this - if we didn't use any fancy classification model, and we simply predicted that **everyone was disease free** - we'd actually get a higher accuracy rate 98%! Unfortunately, now our model doesn't look quite so amazing!

So what can we do? Well, there are a couple of supplementary metrics we can use over and above Classification Accuracy...

Solution: Check other metrics such as Precision, Sensitivity(Recall), and the F1-Score

While we've seen that **Classification Accuracy** can be a good indicator of model performance - we should always dig deeper and analyse both **Precision & Recall** before getting too excited about our accuracy measure, especially in cases where the data is biased heavily towards one of the classes...

Precision

Definition: Of all observations that were predicted, or classified as positive - what proportion actually were positive

In our example: Of all patients we predicted to have the disease - what proportion actually did

We can easily calculate this from our Confusion Matrix by taking the number of True Positives, and dividing that by the sum of True Positives & False Positives

		Actual Result	
		Disease	No Disease
Predicted Result	Disease	1 TRUE POSITIVE	2 FALSE POSITIVE
	No Disease	1 FALSE NEGATIVE	96 TRUE NEGATIVE

$$\frac{TP}{(TP + FP)}$$
$$\frac{1}{(1 + 2)}$$
$$\frac{1}{3}$$
$$33\%$$

Of all patients we *predicted* to have the disease, only 33% actually did! We would have patients thinking they were ill, when they were not!

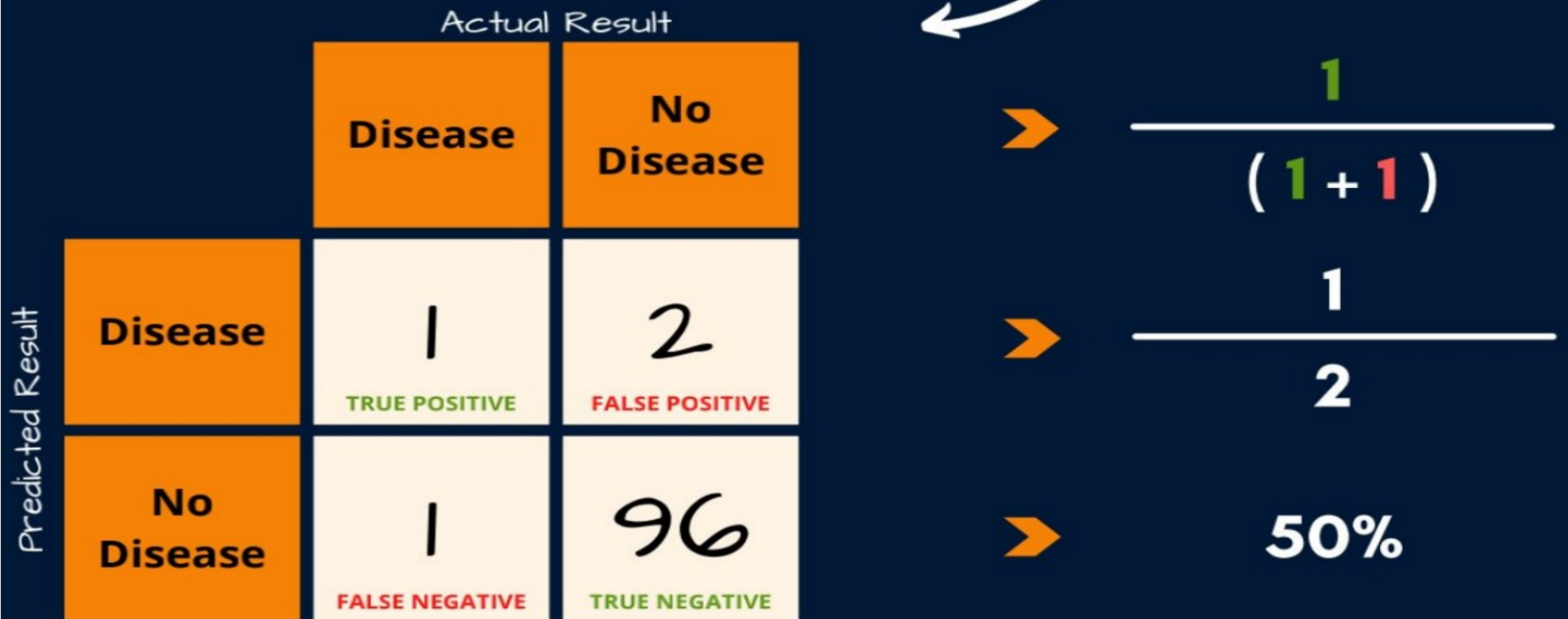
This gives us a whole new way to think about **how good** our model really is!

Recall

Definition: Of all positive observations, how many did we predict as positive

In our example: Of all patients who actually had the disease, how many did we correctly predict

We can easily calculate this from our Confusion Matrix by taking the number of True Positives, and dividing that by the sum of True Positives & False Negatives



Of all patients who *actually* had the disease, our model only predicted 50% of them!

F1-Score

The **F1-Score** is the harmonic mean of **Precision** & **Recall**.

A good F1-Score comes when there is a **balance** between Precision & Recall, **rather than a disparity** between them.

$$\text{F1-Score} = \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})}$$

In our example we had...

Precision = 0.33

Recall = 0.50

$$\rightarrow \frac{2 * (0.33 * 0.5)}{(0.33 + 0.5)}$$

$$\rightarrow \frac{0.33}{0.833}$$

$$\rightarrow \mathbf{0.396}$$

Note

Precision	Recall	Meaning
High	High	The model is differentiating between classes well
High	Low	The model is struggling to detect the class, but when it does it is very trustworthy
Low	High	The model is identifying most of the class, but is also incorrectly including a high number of data points from another class
Low	Low	The model is struggling to differentiate between classes

Main Sources for these slides

- *Michael Sikorski and Andrew Honig, "Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software"; ISBN-10: 1593272901.*
- *Xinwen Fu, "Introduction to Malware Analysis," University of Central Florida*
- *Sam Bowne, "Practical Malware Analysis," City College San Francisco*
- *Abhijit Mohanta and Anoop Saldanha, "Malware Analysis and Detection Engineering: A Comprehensive Approach to Detect and Analyze Modern Malware," ISBN: 1484261925.*

Thank you