All.Net Analyst Report and Newsletter

Welcome to our Analyst Report and Newsletter

Some comments on the Phish Scale[1]

NIST released the Phish scale as a way to measure the quality of personnel in defending against so-called phishing attacks – emails sent to people to get them to do the wrong thing.

It proclaims 2 dimensions for use by those who seek to test and train susceptibility to these acts; 1) the complexity of detection from readily available observables and 2) the act and reports of the acts by their targets.

The basic idea

If you are going to have a training program to reduce human susceptibility to a cognitive attack, you are hopefully going to want metrics to tell whether the program is actually reducing susceptibility and the same metrics perhaps to determine when to stop.

- If you don't measure, you don't know whether what you are doing is working.
- If you keep going when no progress is made, you are wasting time and effort.

The authors folks didn't notice the second one, but they were likely busy trying to get the first one right.

The cues

To detect a cue, you must have clue. The cues in this case are the user observable indicators of phishing, which the NIST publication identified, based on statistics from whenever they were taken, as:

> "the properties of an email that either compel a user to click on a fraudulent link or attachment or alert the user that the email may be a phish".

They provide a list including errors, technical indicators, visual presentation indicators, language and content, and common tactics, and drill down a bit into a table and provide a nice appendix. The idea is that the more cues present, the easier it is to detect, so it takes more clue with less cue.

Premise alignment

This is about the clue of the threat actor… sort of. In essence, it asserts:

> "a measure of how closely an email matches the work roles or responsibilities of an email's recipient or organization. The stronger an email's premise alignment, the more difficult it is to detect as a phish. Inversely, the weaker an email's premise alignment, the easier it is to detect as a phish."

In other words, more attacker clue means more premise alignment. NIST identifies 5 elements to this; 1) Mimics the workplace process or practice, 2) has workplace relevance, 3) aligns with other situations or events (including external to the workplace), 4) engenders concern over consequences of NOT clicking, and 5) you have been warned (my words).

1 https://nvlpubs.nist.gov/nistpubs/TechnicalNotes/NIST.TN.2276.pdf

Scoring – page 2

The scoring of these factors gets a bit tricky, but before I start with that, I think I have provided enough facts to allow you to imagine this article is praise. But you should know me better than that by now. My job is to find the holes and how to exploit them, and figure out how to make it harder to do so. That goes for all defenses, not just technical ones.

Basic problems

The most basic problems here are at the level of the model. The model says more clues = more detectable, better alignment = less detectable. But people are funny that way. I often misspell things and voice input often changes words that I don't catch before a message is sent. AI automated attack generation doesn't have these problems, so the detection scheme will give low cue for high end threat actors, which means the most dangerous ones will get the lowers cue scores and be measured as safe.

I actually provide things for free, like this article, and the use of the term "free" is a trigger for lots of spam filters, as well as hitting the too good to be true category. And yet I never phish (I do send commercial emails however). And the folks who actually send this stuff have indicated that they keep the misspellings and bad grammar because it is consistent with their narrative of not being native speakers. And it works! And of course they can send lots of these emails so they don't care which ones you detect, because they don't have to win very often.

But then the idea here is not to deprecate phishing, only to protect your company, or so the narrative goes. Until we consider that self defense works better when we do it together. United we stand, divided we fall.

There are also measurement problems big time because one set of metrics is all facts (quantitative – sort of) and the other is all judgment (qualitative). But hey, what's a little metrics challenge between friends?

Why not automation?

Now let's be clear. Everything identified as detectable in terms of cue is at least as easily detected by automation as by humans. The premise here is that the human is the last line of defense and that people are supposedly going to pick things up that automation does not. But if you can train a person to detect technical things, you can certainly program a computer to do it. And if you cannot program the computer to do it, how do you expect a person to do it? It's destined to fail.

So you better start by using all the technology you have to counter the fake phishing attacks in your testing program and then fixing your automated detection system until none get through. Which means false positives and loss of commercially valuable emails. But people will do the same thing, and training distrust also has a tendency to lead to employee distrust of the company, turning behavior and increased insider threat. Oh my!

Of course if we use the fake phishing threats to train our automated defenses, and if we do it well, then we will eliminate their effectiveness as a test of our workers. Which means we have to remove our defenses to test the next layer. But the testing and training in less realistic scenarios will train away from the desired actual behaviors. So we will be training and measuring the wrong things, improving the wrong things, and that is not a good idea.

Alignment

I think I have bashed cue enough for now, and so it's off to alignment. Here's the problem with measuring alignment as difficulty. The more realistic the fake is, the harder it is to detect. That's the basic premise of the alignment metric. But it's also the basis of trust. The more aligned with our expectations something is, the more we will trust it. So the seeming outcome of more effective testing with closer alignment is that it will become harder and harder to tell the difference. Which means that if we actually train for this differentiation, we will be throwing sand into the gas tank of the organization.

Suppose the CEO sends you an email asking you to come to a meeting being held right now 2 floors above you. What do you do? Suppose it's your boss?

- Option 1: Ignore it, they never ask you to meetings so it must be a phish.
  - If it is real, you likely just made a (big?) mistake.
- Option 2: Go to the meeting, find out it was a phish (or not)
  - If it is fake, you have wasted time and possibly embarrassed yourself.
- Option 3: Call them up to question it and verify the meeting.
  - It's fine to do once, but if everyone does it every time… (perhaps no more meetings? A good thing? Just saying...)
- Option 4: Call security and report it.
  - See options 1, 2, and 3 above and combine all the bad stuff from each of them. Yes, we all know that security claims they are there to help you, but really?
- Option 5: Something else… lots of things like a standard way to setup meetings...

No right answers here… Anything but Option 2 is a big problem for the organization. And of course if the meeting is 2 days from now and requires a trip to another city, there are coordination activities that might give it away. And if it is a deception it will take even more time and effort and cost more.

The solutions lie elsewhere

Realistically, the use of metrics for anti-phishing programs is a fine thing to try, but be careful what you train for. Inducing suspicion, reducing human performance by creating distrust, disgruntling your workers, and other related things can actually work against effective security. Where is the metric for the bad effects of training?

> "One benefit of a strong and resilient security posture is safeguarding internal and external trust."

Conclusions

The folks at NIST are not fools. And the technical note tries to identify the limitations of the methodology and the need for other methods. But it misses at the most fundamental level the need to understand and measure the overall effect of the program in context. It seeks to measure something and finds a way to do it. And despite its many flaws, it represents some progress. But if you actually try to implement it in an enterprise, you will likely find many problems. And you should try...

