1st March 2013

The Pyramid of Pain

Update 2014-01-17

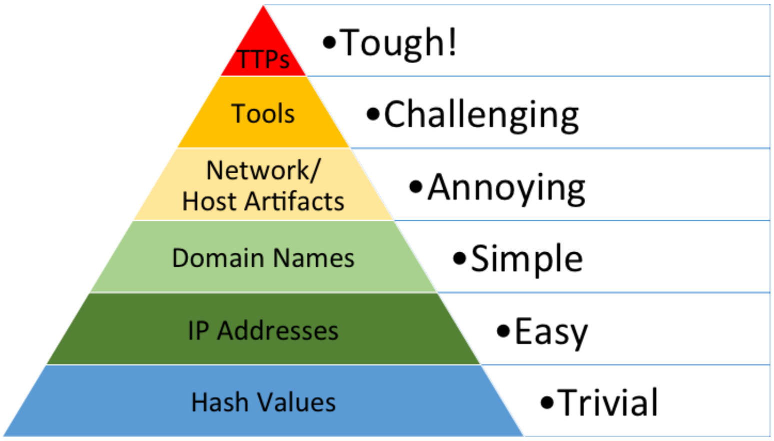
I'm updating this post to include a slightly revised version of the Pyramid. The only real change I made was that I added a new level for hashes. I also updated the text to account for this.

On February 18th, Mandiant [http://mandiant.com/] put a major hole in the APT intelligence dam when they released their APT1 report [http://mandiant.com/apt1] profiling a group commonly referred to as Comment Crew. There followed a small flood of reports from other entities like the Symantec [http://www.symantec.com/connect/blogs/apt1-qa-attacks-comment-crew] (and this [http://www.symantec.com/connect/blogs/apt1-additional-comment-crew-indicators-compromise]) and the DHS/FBI [http://www.us-cert.gov/ncas/current-activity/2013/02/22/Ongoing-Malicious-Cyber-Activity-Against-US-Government-and-Private] . Most of the furor over the APT1 report was regarding it's findings suggesting that APT1 is actually the PLA's Unit 61398. This is solid work, and a real breakthrough in the public policy area, but I was a lot more excited about the detailed technical information included in the reports' seven appendices (not including the video appendix).

After seeing how these indicators were being applied, though, I came to realize something very interesting: **almost no one is using them effectively**.

I put that statement in bold, because it's a little bit of a challenge, and I'm sure it will surprise many readers. The entire point of detecting indicators is to respond to them, and once you can respond to them quickly enough, you have denied the adversary the use of those indicators when they are attacking you. Not all indicators are created equal, though, and some of them are far more valuable than others.

The Pyramid of Pain



[http://4.bp.blogspot.com/-EDLbyYipz_E/UtnWN7fdGcI/AAAAAAAANno/b4UX5wjNdh0/s1600/Pyramid+of+Pain+v2.png]

To illustrate this concept, I have created what I like to call the Pyramid of Pain. This simple diagram shows the relationship between the types of indicators you might use to detect an adversary's activities and how much pain it will cause them when you are able to deny those indicators to them. Let's examine this diagram in more detail.

Types of Indicators

Let's start by simply defining types of indicators make up the pyramid:

1. **Hash Values:** SHA1, MD5 or other similar hashes that correspond to specific suspicious or malicious files. Often used to provide unique references to specific samples of malware or to files involved in an intrusion.
2. **IP Addresses**: It's, um, an IP address. Or maybe a netblock.
3. **Domain Names**: This could be either a domain name itself (e.g., "evil.net") or maybe even a sub- or sub-sub-domain (e.g., "this.is.sooooo.evil.net")
4. **Network Artifacts**: Observables caused by adversary activities on your network. Technically speaking, every byte that flows over your network as a result of the adversary's interaction could be an artifact, but in practice this really means those pieces of the activity that might tend to distinguish malicious activity from that of legitimate users. Typical examples might be URI patterns, C2 information embedded in network protocols, distinctive HTTP User-Agent or SMTP Mailer values, etc.
5. **Host Artifacts**: Observables caused by adversary activities on one or more of your hosts. Again, we focus on things that would tend to distinguish malicious activities from legitimate ones. They could be registry keys or values known to be created by specific pieces of malware, files or directories dropped in certain places or using certain names, names or descriptions or malicious services or almost anything else that's distinctive.
6. **Tools**: Software used by the adversary to accomplish their mission. Mostly this will be things they bring with them, rather than software or commands that may already be installed on the computer. This would include utilities designed to create malicious documents for spearphishing, backdoors used to establish C2 or password crackers or other host-based utilities they may want to use post-compromise.
7. **Tactics, Techniques and Procedures (TTPs)**: How the adversary goes about accomplishing their mission, from reconnaissance all the way through data exfiltration and at every step in between. "Spearphishing" is a common TTP for establishing a presence in the network. "Spearphishing with a trojaned PDF file" or "... with a link to a malicious .SCR file disguised as a ZIP" would be more specific versions. "Dumping cached authentication credentials and reusing them in Pass-the-Hash attacks" would be a TTP. Notice we're not talking about specific tools here, as there are any number of ways of weaponizing a PDF or implementing Pass-the-Hash.

The Pyramid Explained

Now that we have a better idea what each of the indicator types are, let's take a look at the pyramid again. The widest part of the pyramid is colored green, and the pinnacle of the pyramid is red. Both the width and the color are very important in understanding the value of these types of indicators.

Hash Values

Most hash algorithms compute a message digest of the entire input and output a fixed length hash that is unique to the given input. In other words, if the contents of two files varies even by a single bit, the resultant hash values of the two files are entirely different. SHA1 and MD5 are the two most common examples of this type of hash.

On the one hand, hash indicators are the most accurate type of indicator you could hope for. The odds of two different files having the same hash values are so low, you can almost discount this possibility altogether. On the other hand, *any* change to a file, even an inconsequential one like flipping a bit in an unused resource or adding a null to the end, results in a completely different and unrelated hash value. It is so easy for hash values to change, and there are so many of them around, that in many cases it may not even be worth tracking them.

You may also encounter so-called *fuzzy hashes*, which attempt to solve this problem by computing hash values that take into account similarities in the input. In other words, two files with only minor or moderate differences would have fuzzy hash values that are substantially similar, allowing an investigator to note a possible relationship between them. Ssdeep [http://ssdeep.sourceforge.net/] is an example of a tool commonly used to compute fuzzy hashes. Even though these are still hash values, they probably fit better at the "Tools" level of the Pyramid than here, because they are more resistant to change and manipulation. In fact, the most common use for them in DFIR is to identify variants of known tools or malware, in an attempt to try to rectify the shortcomings of more static hashes.

IP Addresses

IP addresses are quite literally the most fundamental indicator. Short of data copied from local hard drive and leaving the front door on a USB key, you pretty much have to have an network connection of some sort in order to carry out an attack, and a connection means IP Addresses. It's at the widest part of the pyramid because there are just so many of them. Any reasonably advanced adversary can change IP addresses whenever it suits them, with very little effort. In some cases, if they are using a anonymous proxy service like Tor or something similar, they may change IPs quite frequently and never even notice or care. That's why IP Addesses are green in the pyramid. If you deny the adversary the use of one of their IPs, they can usually recover without even breaking stride.

Domain Names

One step higher on the pyramid, we have Domain Names (still green, but lighter). These are slightly more of a pain to change, because in order to work, they must be registered, paid for (even if with stolen funds) and hosted somewhere. That said, there are a large number of DNS providers out there with lax registration standards (many of them free), so in practice it's not too hard to change domains. New domains may take anywhere up to a day or two to be visible throughout the Internet, though, so these are slightly harder to change than just IP addresses.

Network & Host Artifacts

Smack in the middle of the pyramid and starting to get into the yellow zone, we have the Network and Host Artifacts. This is the level, at last, where you start to have some negative impact on the adversary. When you can detect and respond to indicators at this level, you cause the attacker to go back to their lab and reconfigure and/or recompile their tools. A great example would be when you find that the attacker's HTTP recon tool uses a distinctive User-Agent string when searching your web content (off by one space or semicolon, for example. Or maybe they just put their name. Don't laugh. This happens!). If you block any requests which present this User-Agent, you force them to go back and spend some time a) figuring out how you detected their recon tool, and b) fixing it. Sure, the fix may be trivial, but at least they had to expend some effort to identify and overcome the obstacle you threw in front of them.

Tools

The next level is labelled "Tools" and is definitely yellow. At this level, we are taking away the adversary's ability to use one or more specific arrows in their quiver. Most likely this happens because we just got so good at detecting the artifacts of their tool in so many different ways that they gave up and had to either find or create a new tool for the same purpose. This is a big win for you, because they have to invest time in research (find an existing tool that has the same capabilities), development (create a new tool **if they are able**) and training (figure out how to use the tool and become proficient with it). You just cost them some real time, especially if you are able to do this across several of their tools.

Some examples of tool indicators might include AV or Yara signatures, if they are able to find variations of the same files even with moderate changes. Network aware tools with a distinctive communication protocol may also fit in this level, where changing the protocol would require substantial rewrites to the original tool. Also, as discussed above, fuzzy hashes would probably fall into this level.

Tactics, Techniques & Procedures

Finally, at the apex are the TTPs. When you detect and respond at this level, you are operating directly on adversary behaviors, **not** against their tools. For example, you are detecting Pass-the-Hash attacks themselves (perhaps by inspecting Windows logs) rather than the tools they use to carry out those attacks. From a pure effectiveness standpoint, this level is your ideal. If you are able to respond to adversary TTPs quickly enough, you force them to do the most time-consuming thing possible: **learn new behaviors**.

Let's think about that some more. If you carry this to the logical extreme, what happens when you are able to do this across a wide variety of the adversary's different TTPs? You give them one of two options:

1. Give up, or
2. **Reinvent themselves from scratch**

If I were the adversary, Option #1 would probably look pretty attractive to me in this situation.

Effective Use of APT1 Indicators

Now that we've covered all that background, we can finally turn our attention back to the APT1 indicators. I said that almost no one was making effective use of them. What did I mean by that, and what constitutes "effective use"?

What I meant was that I see a lot of discussion about the long list of domain names included in Appendix D (and to a lesser extent, the domains and IPs included in the DHS/FBI and Symantec reports as well). Seth Hall [https://github.com/sethhall] of the Bro-IDS [http://www.bro-ids.org/] project even put out a neat Bro module [https://github.com/sethhall/bro-apt1] that you could use to search for those domains in your network traffic.

This is all good and proper, but I haven't seen a lot of discussion centering around the data they provided on the behavior of the Comment Crew tools themselves. Appendix A is a giant data dump of some very good information regarding host and network artifacts for 40+ tools known to be in use by this group.

Emerging Threats [http://emergingthreats.net/] has published a set of Snort signatures that cover many of these, and I'm quite sure many of us have produced our own, but I find the lack of attention to these curious. Maybe the ET rules are already serving everyone's needs and so there's no need to talk about them? More likely, though, I think a lot of organizations have not properly reviewed the reports for detection indicators.

Whenever you receive new intel on an adversary (whether it be APT1/Comment Crew or any other threat actor), review it carefully against the Pyramid of Pain. **For every paragraph, ask yourself "Is there anything here I can use to detect the adversary's activity, and where does this fall on the pyramid?"** Sure, take all those domains and IPs and make use of them if you can, but keep in mind that **the amount of pain you cause an adversary depends on the types of indicators you are able to make use of**, and create your plan accordingly.

Posted 1st March 2013 by DavidJBianco

3 View comments

Matt H January 23, 2015 at 7:16 PM

I prefer to think of it as a pyramid of pain relief! Following these excellent recommendations may be difficult to get going, but once implemented the result should ease any SOC's requirement for remedial action. Preventing incidents can be painful, but responding to them is usually worse!

Reply

Anonymous May 22, 2016 at 2:21 PM

I am regular reader of your blog and no doubt it all stuff is awesome. The best thing about your sharing and posting is that you always provide content that is helpful for both the newbie and experts. Looking for more stuff and tutorials.

Love from Tech Solutions Desk

Reply

Mike Anders December 14, 2016 at 3:24 PM

It is useful to also have a methodology. Taking a Activity Based Intelligence (ABI) approach to evaluating indicators and assessing risks works very well. In fact formalizing things with Object Based Production (OBP) to further enable ABI makes great use of the Pyramid of Pain! AND brings immediate relief. Take two and call me in the morning!

Reply