Active Defense

Do you have something to say about this article?

Visit the *Journal* pages of the ISACA® website *(www.isaca.org/journal)*, find the article and click on the Comments link to share your thoughts.

https://bit.ly/2Ib0huk

Every year, ISACA® conducts its annual Global State of Cybersecurity research.[1] The second part of the survey asks about security practices in the field: what is working and what is not. No need to spoil it for readers (who will have to wait for the report to come out for the specifics), but this year there were interesting findings when respondents were asked about active defense. Specifically, active defense methods are used more frequently than one might think—and they are highly effective when they are used.

What Is Active Defense?

Like many things in security, active defense—as a term—is borrowed from defense terminology. Specifically, it refers to actions taken to deny a position, resource or other advantage to an adversary. In the context of cybersecurity, it refers to measures that can be taken to actively disrupt or interfere with an attacker's campaign against an environment. For example, if a security team were to install a honeypot and load it with juicy-seeming (but fake) documents to actively waste the attacker's time, that would be an active defense strategy.

It bears noting here that I do not mean "hacking back" for the purposes of this discussion.

Ed Moyle
Is a founding partner of the analyst firm Security Curve. Prior that, Moyle was director of thought leadership and research at ISACA. In his nearly 20 years in information security, he has held numerous positions including senior strategist with Savvis, senior manager with CTG's global security practice, vice president and information security officer for Merrill Lynch Investment Managers. Moyle is coauthor of *Cryptographic Libraries for Developers* and a frequent contributor to the information security industry as an author, public speaker and analyst.

Sometimes, there is confusion about the distinction between "active defense" and so-called "hack back" strategies. In my opinion, active defense is designed to disrupt an attacker's activity through minimally-invasive and clearly delineated strategies. By contrast, "hacking back"—for example, by attempting to scan, penetrate or gain entry into an attacker's environment—is (again, in my opinion) both ethically and potentially legally problematic. So, if there was any confusion before, let it be clear now that this discussion refers throughout to active defense strategies and not attempting to "hack the hackers." Specifically, it refers to things that either:

- Waste the attacker's time
- Trap and contain attacks
- Alert security teams to attacker activity so it can be monitored
- Help with attribution and discovery

There are a few reasons why active defense can be a particularly useful and effective strategy. First and foremost, it can help to disrupt an attacker's campaign. As we know from looking at adversary activity as a life cycle (i.e., as a "kill chain" that starts with reconnaissance, proceeds to infiltration and lateral movement, and ends with exfiltration or some other equally undesirable outcome) any interruption in the attacker's ability to proceed from phase to phase can cause the overall campaign to fail. Likewise, there is "dwell time" of which to be conscious; the campaign has a window of time between when it is initiated and when it is discovered. Anything that can frustrate attackers' ability to realize their outcome increases likelihood that the attack can be detected and stopped within that window before the attackers are successful.

In addition to that, though, active defense can also assist with attribution. This is particularly useful from a law enforcement point of view; for example, it can support criminal proceedings against someone or warn others of an attacker campaign or tradecraft. In other words, if an organization identifies or can establish who is responsible for attempting an attack, it is useful information that can be passed along to law enforcement.

Tools

With this in mind, there a few free tools to consider when looking to get familiar and just play around with active defense strategies. Like any security tool, practitioners do not want to just field these willy-nilly, but instead, for a production deployment, with care and strategic foresight. That said, it is always helpful as a starting point to kick the tires and gain familiarity.

It probably goes without saying that there is a grey area here, so speaking with internal counsel about any usage scenario is important before you field tools such as these (better safe than sorry). That said, active defense can be—and is—a useful strategy to help support a security program.

Endnote

1 ISACA, State of Cybersecurity 2018, *www.isaca.org/state-of-cybersecurity-2018*

1

One of the better starting points is a honeypot. A relatively versatile one to tinker with—and one that is well documented from a usage standpoint—is OpenCanary *(https://github.com/thinkst/opencanary)*. OpenCanary makes a good starting point because, conceptually, it is fairly simple: It runs services and triggers when someone connects to them. The services in question are designed to emulate a particular device configuration (e.g., a Windows or Linux server).

2

There are, as one might imagine, literally dozens of other open-source honeypot options to choose from and, depending on what type of service or environment practitioners want to emulate, there are plenty of choices available (a helpful list can be found here: *https://github.com/paralax/awesome-honeypots#honeypots*). It is, of course, useful to select a honeypot that resonates with an environment. For example, if one is running a 100 percent Windows shop, a Secure Shell (SSH) honeypot designed to mimic a Linux web server might seem out of place. Ideally, one that will blend in with services already in use should be selected. Another option to consider is WebTrap (*https://github.com/IllusiveNetworks-Labs/WebTrap*).

WebTrap is small, recent and very targeted. It lets the practitioner mirror an existing web page (e.g., a corporate information portal or project page) and alert (e.g., via a syslog event) when someone interacts with it. That said, any web server can be customized for this purpose by mirroring an internal page and setting up custom reporting when it is accessed.

3

Taking the honeypot concept one step further is HoneyBadger v2 *(https://github.com/lanmaster53/honeybadger)*. HoneyBadger includes a framework for geolocation—helping to pinpoint where the attacker is located. This can help bolster attribution capability by providing information about the location from which the remote attacker is coming. Used in combination with a tool such as Molehunt *(https://github.com/Prometheaninfosec/Molehunt)*, one can get a fairly clear picture of who is attacking and the attacker's location since Molehunt allows the user to create documents that, when opened, let security teams know about it. One might, for example, deliberately allow a document to be exfiltrated and, together with HoneyBadger, glean attribution-relevant information about the person opening it. Again, this tool is minimally intrusive and solely focused on information gathering and assisting law enforcement.

4

The last approach discussed here is the Browser Exploitation Framework (BeEF) *(https://beefproject.com/)*. BeEF is arguably more of a penetration testing tool than an active defense tool, but it bears mentioning because it can, in certain circumstances, be used to support active defense as well. First, it is important to note that it is imperative to use this tool in a lawful way. If there are doubts about whether an approach is lawful, it should be discussed with the legal team. If a user cannot determine if the planned usage is lawful, it is better to err on the side of caution and forego this one for now. So, caveat complete, BeEF allows the practitioner to "hook" a remote user's browser when the user navigates to a page controlled by the practitioner. After so doing, the browser can be used to collect information, track the remote user's activity and, in certain circumstances, map the remote network from which they are connecting. Per the caveat stated previously, information-gathering techniques to support attribution (i.e., the location from which someone is connecting) is probably lawful (once again, the legal team should be consulted) whereas other techniques (e.g., mapping the user's remote network) probably are not without the user's permission. In a penetration testing context, this means a practitioner can use a browser as a launching point to infiltrate an environment. From an active defense point of view, though, this could mean soliciting attribution information.