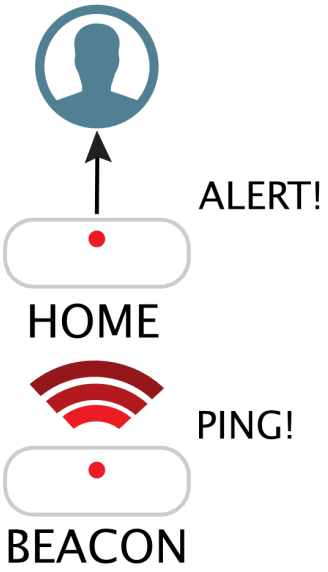
Experiences with the Beacon deployments

In this whitepaper about SensorFu Beacon:

- **What SensorFu Beacon offers for network segregation monitoring?**
- **How and why network segregation can fail?**
- **What we have learned doing real world Beacon deployments?**

SensorFu Beacon is a software product that detects new network leak paths from isolated networks or network segments. These leak paths can be a result of human error or malice, and they may violate your security policies or contractual obligations. Product consists of two parts: *Beacons* that continuously look for new network leak paths by the means of active network-fu, and *Home* that listens for successful escapes and is used to create and manage Beacons.



A brief history of Beacon

We have delivered our SensorFu Beacons into a real production networks. To celebrate this milestone of real deployments, we are writing about the lessons learned and experiences gained while working with our early adopters.

Back in early 2017 when we began the journey, our goal was to start by solving a nagging problem that had become all too familiar over the years of performing network security audits: Supposedly segregated networks were everywhere, but frequently that was all they were, supposedly segregated. One of the key observations, besides leaky segregation, was that often the failures were rather simple from a technical standpoint. Complexity arose from the multitude of potential points of failure: A cable mistakenly plugged into a wrong switch port, misconfigured VLAN tagging, multihoming network device, too promiscuous firewall rules and so on. Another, glaringly obvious problem was that while audits did help in fixing these issues, they are by their very nature a point checks in time. A lot can happen during a year, for example, in the case of annual audits.

And so the idea was born: What if we could create an automated audit solution for testing network segregation policy. One that would cover those multiple and diverse points of failure. It would be always on, providing immediate feedback once the leak occurs, instead of the user having to wait until the next audit engagement. Beacon seemed like an apt name for this simple, yet powerful, solution. Although, unlike with the seafarers, all is well as long as you *don't see* SensorFu Beacon.

Drivers for setting up the Beacon

On the surface, drivers for Beacon adoption are straightforward: Regulatory requirements for network isolation, internal and external policies for safeguarding the critical data and communications, verifying the segregation of development networks for different projects and so on. We've heard all of these, but it's interesting to look at some of the less articulated reasons that have emerged during the customer engagements over the past year.

Customers acknowledge that human errors can happen before, during and after the deployment of network technologies. While this is well known for anyone in the security industry, it's refreshing to see that this simple reason can be openly discussed nowadays.

Our next motivation is related to trend in industry where increasing amount of infrastructure is being outsourced. In many cases, for example with hosted cloud services, customer's only option has been to trust the vendor assurances. As Beacon doesn't require special permissions, network access or hardware deployment, "trust but verify" is finally possible in live networks in a lightweight manner.

> *Speed matters; the expectations of technology are shifting so fast that it's difficult for a company to keep up.*
>
> *Jamie Smith, CIO, ServiceMaster (from cio.com)*

New technologies organizations are deploying, and speed of change, pose challenges similar to the above examples. Configuration options will be new or dissimilar to those of previous systems, which gets us back to the increased likelihood of human error. Testing options in live networks are limited, so trusting the vendor has often been the only verification method for the technology itself. In this scenario, Beacon offers always on continuous verification that will run in the background while the deployment progresses.

Beacon deployment

This chapter will be really short. One of our core design goals since the early planning stage was that Beacons should be very easy to deploy and spawn throughout the customer organization. This works well with our second design goal: Help our customers find and fix one real problem in 300 places instead of finding 300 potential problems in one place. After the Home deployment, we are hardly hearing a peep from customers while they are deploying Beacons. Unnerved by the silence and checking the status, we're hearing that, yes, Beacons are happily humming along. We must have succeeded on ease of deployment goal!

Beacon sighted — our findings

As with any security solution, we're only seeing a fraction of results produced in customer environments. These examples are based on observations during cases where we have been working in close cooperation with the customer. Nevertheless, they offer a glimpse on common points of failure and what to pay attention to when sealing the networks that should be watertight. The two most common failure modes we've been seeing are related to firewall configurations: too permissive "allow" rules, and IPv4 vs. IPv6 setup.

Permissive "allow" rules can happen for example in cases where different teams manage different assets: One team is carefully taking care of racks full of servers while another team is managing networks and firewalling. Server team asks for firewall ports to be opened and firewall is adjusted. Time passes by and it's time to refresh. Old servers are decommissioned and some are replaced with new ones, maybe even with new role. New servers runs happily with the current firewall rules, but did anyone check if the rules are too permissive? And network team is happily monitoring the flows, unknown what changes server team has done.

While IPv4 has been around for ages and organizations have long experience in tightening up IPv4 networks, IPv6 seems to be getting less attention. IPv6 is a lot more than just increased version number. Many old technologies are replaced with new IPv6 solutions. For example the address resolution protocol (ARP) has been replaced with neighbor discovery protocol (NDP) running on top of ICMPv6. IPv6 also comes with the built-in zero configuration support making it a breeze to get clients connected into network. Old advice has been to disable IPv6, but in reality it's almost impossible. With bring your own device (BYOD) policies, IoT and everything everywhere configured to IPv6 enabled and zero configuration working as intended, mistakes can happen. Learning this new technology takes time and practise, and there's always the feeling *"did I get it right?"*

An interesting observation has been made with organizations running critical networks with multi-layered defences. In these types of environments, we have seen several cases where some of the layers fail. This could in worst case lead to potential threat moving laterally inside networks moving from Internet to office network to industrial control systems network gaining foothold inside the most protected systems. Thinking about security as a multi-layer defence (onion model) definitely pays off and by observing Beacon propagation inside network layers allows sealing the gaps in continuous manner.

Our last example is related to multihoming network assets. From early on, we expected to see unobvious leaks with multihoming as a root cause. The Ethernet Broadcast escapes provided an opportunity to test this assumption. Most common multihoming devices are firewalls and proxies, where it's part of their function, followed by servers where multihoming may be implemented by design or by accident, and workstations where it's mostly unintentional.

For the purposes of this article, let's cut to the chase and discuss one particular case. In this network we were testing, the culprit turned out to be a specialized embedded device where multihoming fell way beyond device's job description — yet it was merrily forwarding Beacon onwards. Flying under the radar, packets ended up in the operator's transit network and were caught at the perimeter of another isolated segment. That was a sneaky one, and a good real world test for Ethernet Broadcast escapes.

Conclusion

It has been a rewarding ride for us demonstrating that Beacon works as intended in real customer environments and creates value for them. Working with the customers has been invaluable in learning more about their issues and shaping the direction for future development of the product. We have a pool of ideas and customer feedback helps us prioritize and schedule what we should focus on next. We are excited to continue the journey and expand the capabilities of the Beacon!

Related articles

Why Ethernet Broadcast Escape Tests Matter
https://medium.com/sensorfu/6d17924b5233

DNS Servers in Isolated Networks — Will they Leak?
https://medium.com/sensorfu/d7f04fc05884

Special Privileges: No thanks!
https://medium.com/sensorfu/7d5a723776b8