What Is Critical Infrastructure – and How Can Attacks Against It Be Stopped?

Introduction

While it sounds abstract, critical infrastructure can be readily defined:
It's where your critical applications and processes run.

Most IT environments deploy some level of defense in depth to provide protection or worst-case firefighting capability. Stating with limiting user access to only certain applications, limiting network connectivity, and finally using some form of agent-based protection used to try and stop traditional attacks or used for IR activities during an attack to try and limit the spread. While network security is important – for more than a decade, next-generation firewalling along with IDS/IPS have been known as only a first line of defense – it's truly nothing more. Today's endpoint solutions are necessary, but not sufficient to stop anything but today's most basic attacks. Bottom line: it's not working against today's sophisticated attacks.

How big is the problem? From our count since the beginning of 2021, there have been over 700 attacks that have been identified impacting critical infrastructure applications that go right through these defenses.

What is needed to stop these attacks and how do we figure this out?

First rule: We cannot properly stop what we don't understand.
Second rule: It must be easy to stop what needs to be stopped.
We'll delve these rules in greater detail as we look at each option.

Let's tackle understanding the problem in steps:

Applications and processes run on devices.

It is not only where they run that needs to be protected, but also how they run. To explain this further, let's step back to basics on the types of existing attacks that can appear within a private infrastructure environment.

Simple attacks:

These usually come from the loss of key credentials. The means to stop these have been

known for over a decade. It is straightforward to apply techniques to limit user access to certain applications with limited access passwords, and then use proper multifactor authentication programs to confirm it is the user, not an imposter. This stops 30-40% of the application attacks out there. Authentication solutions exist and are well-known. Microsoft Active Directory provides role-based authentication and can leverage multifactor authentication programs with its applications. Most critical application vendors also do both or allow third-party tools such as Active Directory, CyberArk, or Beyond Trust, to name a few, to perform Password Access Management.

What about the rest of the attacks? We're talking about those where code-level or process-level attacks target the most vulnerable and accessible devices on which your critical applications run.

There are three types of code/process level attacks.

1) **Foreign applications appear and activate**: These typically take the form of malware, including a subset known as ransomware. These attacks do tremendous harm and can be the launching point for a sophisticated intrusion if they can open a backdoor by appearing to be legitimate application traffic, using allowed protocols and ports through the firewalls. Once a production environment has been breached – even if it is segmented – malware can spread to all production devices in a segment, compromising those devices, often shutting them down, destroying critical data or even wiping them clean.

2) **Adulteration of legitimate production applications occurs**: This typically happens when an application vulnerability is exploited. Known vulnerabilities are published as CVEs (Common Vulnerabilities and Exposures). This informs people that a patch is needed and tracks when the affected vendor makes one available, which can take time. It is also useful for attackers, as now even the average hacker knows how to attack your applications. Only a small fraction of CVEs reported are patched each year. Critical application vendors may take up to six months to get a verified patch out to you. Of course, you should assess the patch in a lab just to be sure it does not impact something else you have running alongside. This is a serious flaw in the construct of continuous patching to provide meaningful protection.

3) **It is your OS that's being attacked**: OSes are built to run processes called from the applications. To date, it's been difficult to detect attacks on these processes as malicious, and even harder to stop them. They have, up until now, been mostly used by nation-state backed hackers and well-funded cybercriminals. You may hear about living off the land (LOTL) attacks or other strange names. Sometimes they are referred to as part of an advanced persistent threat (APT) – and again, until now, it has been hard to find them, let alone stop them until well after the fact.

You have two choices to stop the attacks.

These are not mutually exclusive. In the right combination, you can do a lot less of the first with a better version of the second – saving money, time, while significantly reducing risk.

1) **Prevent the attacks from landing:** This is where most of the effort has been spent over three decades. That should raise a flag. The theory is, if the attackers cannot reach into these devices, they can't launch the attacks. However, the best network-based defenses only limit the "paths" in. They cannot block them all and still be able to run certain updates or other monitoring processes and services. Some also look for legacy signature detectable malware by sorting through the network traffic. This is expensive and it is no longer effective (more on that below).

   Some networks are air-gapped, meaning no wired connectivity to the outside world. These have become rarer in the drive for digital transformation, productivity, and the well-meaning desire for analytics and visibility, but they still exist in much of the OT world. In normal day-to day-activity, however, humans walk around the firewalls and network air gaps (non-networked paths in). These humans must perform updates to fix applications, or improve their performance, or do other maintenance tasks. Suppliers – application vendors, equipment vendors, third-party system integrators – and even well-meaning internal staff can unwittingly bring the problems in with them.

   This first approach can limit the chances of attack. However, it does not come close to eliminating them. Worse, these approaches can be complicated to design and get to work properly. They are resource-intensive to plan, deploy, and tune. These are often over-sold, giving an illusion of safety. In actuality, they are being bypassed by sophisticated attackers.

   This is about defense in-depth. Let's look at the next step.

2) **Stopping the attacks once they land:** This requires something on the devices where the applications run. This is some form of application that watches what lands. There are four types – we will focus on automated versions because these tools must run without human intervention to be effective at stopping attacks before harm is done.

   a. AV & NGAV

   These are anti-virus (AV) applications. AV, which appeared two decades ago, calculates a value (hash or more commonly known as a signature) for malware files. They can be computationally heavy when running. They worked well until a decade ago, when malware began to become polymorphic, meaning that it could modify itself slightly with each deployment to yield a unique signature calculation when

checked. It is still out there, included with many EDR tools, and it is also typically included with network based NGFWs and IDS/IPS tools. However, they don't find much of any of the most harmful recent malware or ransomware.

NGAV or Next-Gen AV – today's state-of-the-art – watches the attack behaviors (patterns of the attacks), identifying them after they do bad things to the first devices on which these third-party tools are installed. The tools capture the pattern of attack, which is then shipped to their cloud for analysis, where they create an Indicator of Compromise (IoC) based on the specific attack pattern seen and then send down a block for it to interrupt the attack from doing harm. These types of malware are now called "zero days" since they have not been seen before. This does not mean that they were actually stopped on day zero, as it may take days to weeks for that – not great if you were caught during the initial appearances that first week or so. Note this works primarily for malware - the other two types of attacks on the applications and the OS' processes are not stopped this way.

Ideally, these zero-day malware/ransomware attacks would all be stopped on day zero, but this is currently not possible. One of the challenges is there tends to be a new set of IoCs for each attack, along with signatures for older attacks. There are too many to load and run efficiently on devices being protected in many circumstances. A concentration of the recent and most frequent attack identifying IoCs and blocks is therefore kept in each device and shuttled back and forth from the vendors cloud if older attacks start to reappear. This is not ideal, as it requires cloud connectivity and continuous updates. One of the new challenges is that the dawn of generative artificial intelligence (AI) is now being talked about as ushering in new ways attackers can vary the patterns of attack quickly – changing the IoCs on the fly, which will make it much harder to stop these dynamic attacks with today's state-of-the-art methodology.

b. Endpoint Detection Response (EDR)

The NGAV approaches often recommend being run in parallel with EDR solutions, which can provide additional detection of attacks after they become active. Typically, this is mostly a tool for allowing further human investigation and remediation to be done on the device hosting the EDR. This model assumes IT environments with help from third parties or the EDR tool vendors themselves from their SOC (Security Operations Center) with dedicated staff to investigate and take action. This requires access by these teams through your network defenses and is human-driven, which is expensive and takes time – and in the end, remediation

actions may be too late to do anything but clean up the device.

c. Application Allowlisting

This was an interesting concept about a decade ago. The idea is to only allow known, approved applications to run. An application is typically identified by its certificate and if it is on the "allowlist," it is allowed to run. The problem is that most of these allowlisting applications only examine an application at application boot-up. Most application allowlisting solutions just run at system reboot. It is expected that the operator will be watching when a device is booting up and catch something that is not running but should be. Some solutions do also run in the background to detect less frequently used applications and flag/kill them. Either way, they typically need tuning, meaning that operators need to be watching until everything is steady state. They can be effective in environments where new applications aren't frequently introduced and where there is a limited set of applications. It is also important that application updates are controlled to only occur during certain maintenance windows.

Application allowlisting is a good way to stop new applications that appear that are not approved. It is also a good way to kill application updates; these must be approved. In locked-down environments, this approach allows enforcement of policy to control what is allowed to be updated and run in a production environment. It is, however, more problematic in general-purpose environments that employ continuous patching, which will result in the updated applications being blocked when those applications reboot after the patch. While excellent at stopping all forms of file-based malware and ransomware before they can execute, they do little to stop sophisticated attack techniques used by nation-state and cybercriminal attackers to exploit application vulnerabilities and/or the OS's processes.

Because of the complexity in setting up and tuning most of these applications, all but the most highly trained and staffed production environments eschew traditional application allowlisting.

d. Application-Level Zero-Trust

This is a novel approach devised for this decade, where the industry has seen a rapid rise in the number of sophisticated attacks.

Spurred on by their successes – nation-state and cybercrime syndicate-backed hackers have invested heavily in building out multi-stage, automated attack kits to infiltrate and gain control or simply wipe-out critical assets. Starting with the famous

SolarWinds (Sunburst) sophisticated supply chain attack in early 2021 that bypassed all the above-mentioned defenses – and went completely undetected in over 90 large enterprises – there have been hundreds of similar supply chain-enabled attacks.

This zero-trust approach was devised to create a more generic approach to stop sophisticated attacks and all other forms of zero-day malware attacks automatically out-of-the-box. The goal is broad: To stop any of the three code/process-level attacks on the system detailed above and solve the shortcomings of the prior generation of approaches. Solution requirements included:

- Stop the three forms of attacks before harm is done.
- Do so fully automated out of the box.
- Do so without requiring updates to stop each new form of attack.
- Work on legacy operating systems.
- Work in fully air-gapped environments and maintain efficacy.
- Protect at the OS as well as the application level.

To do this it must:

1) Keep all undesired applications from running – any form of malware appearing on day zero must be stopped before it can do harm.
2) Generically keep production (desired) applications from being adulterated – block code-level exploits and vulnerabilities.
3) Generically protect the system's OS from having rogue processes run.
4) Allow the production applications to run without performance impact.

Note: the word "generically" is used above because that is an ideal way to do this – it may not be the only way, but doing so provides the advantages of stopping more attacks with a simple-to-implement approach and doing so in challenging IT and OT environments.

Since only one known (to us) solution today fits all these requirements, we will get specific about how our patented AZT PROTECT™ solution approach meets the requirements. It was designed specifically to protect critical production applications and the OS platforms on which they run.

Approach:

ARIA AZT PROTECT deploys as a kernel level driver – attaching at ring zero. Why is this important? To watch the calls into the OS and the processes running, you must have this

proximity to the kernel. In other words, it helps to be in the path of the bad guys as they try to exploit the system. This makes it hard for the hackers to hide their actions. It also helps to ensure that the protection provided does not get bypassed. Yes, that is a real risk – case in point: at the US Senate hearings on February 23, 2021, four of the leading security providers, including the current endpoint protection leader, revealed this happened to their solutions during the later stages of the SolarWinds supply chain attack.

Most importantly though – the way ARIA AZT PROTECT works is that it watches how applications and processes execute. It does so with a patented approach that continuously watches the memory in use on the platform for each active application. It identifies the application by its binary and memory footprint to build a patented immutable ID – known as a TrustID. Any substantive change to the application while running will change the TrustID's calculation.

The system is protected in three ways:

First, applications can be locked down to just run those the administrator has approved. This can stop all others that appear from running, including malware/ransomware or anything else undesired. Optionally, in a slightly looser mode, it allows updates for the trusted applications to be accepted after the certification is checked as both legitimate and up to date. This can be further extended if desired, to self-signed certificates for in-house applications. Note that the process to do all of this is also patented.

Second, even without enabling application-level lockdown, if a running application gets adulterated, such as through some form of code injection from an intruder, the TrustID will change, resulting in the altered code being blocked from running.

Third, stopping sophisticated attack techniques. Nation-state-backed attackers as well as those hired by well-funded crime syndicates use certain techniques to get by today's best defenses once they have some form of access to the system. We will call these LOTL techniques, but it goes beyond just those. Overall, there are about a dozen unique attack techniques that have not changed significantly in the past seven years of research. Going into each in detail is not practical. The important point is that if you can generically detect the technique – such as an unexpected read or write-to buffers, buffer overruns, heap spray, malicious scripts , shell code, and various unattached OS processes being called upon to run, as well as application privilege escalations, in addition to new or altered applications – you can detect and stop just about all the tools of such sophisticated attackers. ARIA AZT PROTECT does just that.

Proof: We have documented that in six well-known attacks seen over the last three years,

ARIA AZT PROTECT will stop before harm is done. From the SolarWinds, Sandworm, MOVEit, and PoolParty attacks to the more recent Volt Typhoon and UnitedHealth Group Optum attacks, each was different, but all were alarming, as the best tools available allowed these attacks to succeed. See our blogs for more details on each.

In summary, these prominent attacks cover all three of the attack types discussed above.

In two cases – SolarWinds and Sandworm – multiple types of advanced attack techniques appeared during the lifespan of each attack, giving us plenty of opportunity to stop the attacks. SolarWinds started with lightweight malware hidden in their ORION software driver, which called home, pulling in more code and thus launched various hidden attack processes to give complete control to the intruders. Sandworm did the opposite once they gained access. They ran hidden attack processes to exploit critical applications and then ran privilege escalations to get to system-level control. Once done turning off critical processes, such as shutting down the power at utilities, they then launched wiper code – which AZT would prevent – to make it extremely difficult to replace and get the power back on.

PoolParty, revealed in December 2023, was an industry shocker as it illustrated how the OS' thread pools could easily be used to attack the system. The industry leaders in cybersecurity had never imagined this attack vector before. Researchers presenting at Black Hat UK demonstrated they had created eight attacks that went undetected and unstopped by all the leaders of Gartner's latest Endpoint Protection Magic Quadrant solutions. These results were presented to the solution providers who then needed to develop updates to protect from this style of attack – a slow process. The researchers made the point that what was considered best-in-breed protection in last decade is falling short in this decade. ARIA – using a six-month-old release of AZT PROTECT, deployed and protecting customers since July 2023 – stopped all eight attacks before they could get started, out of the box, with no updates required.

The Volt Typhoon attack, from what is known, can be stopped as it tries to run unattached code and escalate privileges. The UnitedHealth Group Optum attack investigation is still ongoing, but from our analysis, AZT PROTECT could have stopped this attack immediately when malicious code was launched. The current industry leaders need to get a sample of the code to see if they can develop a block that can be downloaded. In this case, a generic approach that does not need updates is obviously better.

The key takeaway is that the generic approach used by AZT PROTECT stopped all these attacks. Why is that important? It means we do not need to determine an attack's IoC or have

previously seen the attack at all to block it. This means we can deploy our agent and it will just keep working to stop attacks without updates. It can run fully air-gapped forever and be effective. No need for suspect code to be sent to the cloud for analysis or daily updates to be downloaded, without testing, from who knows where.

Other attributes of interest

We built ARIA AZT PROTECT to help run production applications in a variety of environments. OT environments such as in manufacturing may be running on out of support systems. There is no need to rip and replace. We run on Windows platforms from XP onward, as well as Enterprise Linux, including specialized embedded versions of these OSes/distributions, and for both X86 ARM core platforms. Further, we optimized the agents to use as little as 1 CPU core and a minimal amount of system memory and disk so as not to impact application performance.

Customers have successfully deployed our solution site-wide without us being involved. Entire factories have been brought up, automatically and fully protected in well less than a day. The system runs completely autonomously, so there is no need to have trained staff or a SOC to get involved with stopping the attacks. Critically, ARIA AZT PROTECT provides evidence that can help organizations determine if they need to file a Form 8-K with the SEC to report a material, production-impacting breach. In such cases, the evidence will show that the attacks on critical processes were stopped before there was an impact to operation.

Summary

Critical infrastructure – it is about the applications. It is about proper defense, in depth, in the right measure. Doing it right does not mean overdoing it. You can cut back on network protection spend that more than pays for the active defenses you need to run where your critical/vulnerable applications and processes run. Complex systems take prolonged periods to plan, prototype, deploy, and provision – we have heard of projects that span years. This is expensive and also risky – it only partially solves the problem and takes dedicated staff, to achieve the maximum level of protection. To make matters worse, attackers are launching increasingly sophisticated attacks with each passing month, in higher volume, thanks to an assist from generative AI-enabled attack kits. It is time to take the next step in protection, rather than deploying solutions designed for attacks from the last decade.

To find out more, schedule a demo with an ARIA expert or check out our ARIA AZT PROTECT webpage, including blogs and case studies.