SECURITY PAPER

How Do You Disrupt The Attackers’ Patterns

How can you disrupt the attackers' patterns?

In parts 1 and 2 of this series of papers, I have discussed a possible path for a malicious actor to find and attack information assets and shown how simple it is to perform portions of these tasks with common and freely available tools. In this last of these articles, I will explain how the application of NetFoundry zero trust technology can disrupt these patterns and protect your information and business from being compromised with minimal friction to your users. All three papers are available from the NetFoundry web site.

Zero Trust Network Access

Zero Trust Network Access (ZTNA) describes a set of controls and configurations that takes security to the next level, to deal with the changes and evolutions of networks today. There are many definitions of ZTNA. The USA National Institute of Standards and Technology (NIST) publication 800-207 provides a very good one, with multiple architectures, and other information.

Zero trust architectures advocate that we can no longer trust access just because it is "inside" our network. Today, our networks are more diverse and distributed than ever before, extending into multiple clouds, multiple geographies, and must be accessible by remote users connecting from anywhere over any connection (often unsecure). All of this makes maintaining the security of network operations of a business very difficult. Managing updating/patching and 24x7 monitoring of the security of all our information assets and legacy network systems is complex. Complexity increases the chance of oversights and mistakes that can be exploited by attackers.

NetFoundry’s Implementation of Zero Trust Network Access

NetFoundry's zero trust platform, delivered as a service, allows you to make your applications and infrastructure "dark". That is, the network does not respond to any traffic sent to it that isn't previously authenticated and authorized to access the network or applications and assets on the network. This stops malicious attack patterns before they can start. You cannot compromise something you cannot see. The only users or systems that can reach the network are those you have already securely identified and designated. This is not done by IP address or another item that is difficult to manage, but by digital identity, backed by X.509 certificates and with key exchange based mutual authentication.

A NetFoundry network instance gives our customers an easily configurable zero trust global software-defined network. Network endpoints are software based and access to the NetFoundry network instance is policy based and integrates posture checking for devices. In the best and most evolved cases, NetFoundry zero trust is embedded directly into applications, using our open-source software development kits (SDKs).

Not all scenarios readily require or easily support software-embedded endpoints. The software in question may not belong to the customer, there may be a development project timeframe or software life cycle scenario that creates hurdles. In these cases, NetFoundry has prebuilt applications referred to as tunnelers (Linux), or Desktop Edge applications. These software agents create customer access to their NetFoundry network instance without requiring changes to applications.

How does NetFoundry make your network go dark and inaccessible from inbound connections from the Internet? Information resources on NetFoundry networks are protected by the nature of outbound only, meet me connectivity. With NetFoundry, the zero trust network instance is instantiated, via NetFoundry or customer hosted Edge Routers. Services are then configured. Services are your information network assets, including hosts, applications, and protocol ports.

Endpoints are created and given access to all or groups of services via a NetFoundry AppWAN (Application WAN). AppWANs are dedicated highly secure micro segmented connections configured with least privileged access and specific policy based connectivity capabilities. The hosting application, whether it is an embedded SDK, or one of our prebuilt applications, will connect to the network controller, and retrieve information to access the network.

Similar to the authentication that occurs when you use your bank's or other website, NetFoundry certificates on the server are verified to ensure you are communicating with the correct device. NetFoundry goes further, unlike most secure websites. NetFoundry authentication is also reciprocated and carried out in the reverse direction to the endpoint. The NetFoundry network controller verifies the endpoint's certificate to ensure that it is an authorized node – BEFORE the endpoint is allowed to connect. If this initial authorization fails, or if it fails at any time after an initial authorization, network access is revoked.

Since the connection from the endpoints is always outward only toward the network, there are no open inbound ports, so all incoming connections are simply not possible. This is one of the most key items in combatting ransomware; if your network assets are not accessible from the internet at large, then you have greatly reduced the attack surface of your network.

IP Scanning occurs 24 hours a day, every day. An internet reachable node will receive packets from thousands of systems a day, and many more connections if any ports are exposed, probing for weaknesses. Networks have used architectures including firewalls and VPN servers for decades to provide remote access to users while trying to protect systems from scanning and subsequent malicious access attempts. Unfortunately, time has shown that VPNs have serious weaknesses. The requirements to support today’s highly distributed network architecture have rendered traditional VPN and Firewall insufficient.

Implementing ZTNA Networks – Three Scenarios

There are three scenarios for implement NetFoundry ZTNA networks to increase "zero trustiness". The first scenario leverages a gateway device, also called an enclave design. This architecture places a software agent on a node attached to a network with internet access. Services are configured, and the local nodes may be reached to and from the Internet by their local IP address or resolved name. This architecture appeals to some as it is very much like traditional routing. However, this is the least secure architecture. In this scenario, while the network traffic is encrypted, and it is absolutely coming from an authenticated and authorized node, the originator or server is not verified to the network, only the gateway is.

That means that we lose context and control, since other unknown devices may access the gateways. If this scenario is deployed in production, the network is not true zero trust, and the actual goals of the implementation should be measured with an understanding of these vulnerabilities. As a testing architecture, a step in implementation, or some specialized architectures, this design works well and is fully supported. NetFoundry believes this is not true zero trust, and do not recommend this model except when it is very well-considered against specific and identifiable goals – and an understanding of risks are clearly recognized.

The second, and most common initial deployment architecture is host-based. In a host-based deployment, the NetFoundry tunneler or desktop edge software is deployed onto every system. This may be done manually in small deployments or using automation systems such as Ansible, Terraform, or others in use already in larger networks. The host-based deployment allows a much finer granularity of control, as well as understanding the context of collected data. Rather than workstations in a branch office all accessing a common gateway, each device has a separate AppWAN connection to the network. Combined with endpoint security controls, such as passwords, MFA, lockout timers, etc. the network administrator can have a high degree of confidence that the node accessing a network service is being used by the assigned and authenticated user. This architecture gives fine-grained control over what users can access what services, (when and how) and provides strong certificate based identification and authentication of those users and services, which is the heart of zero trust.

The third and most secure implementation is software embedded. This architecture is unique to NetFoundry. Rather than only providing a client, we provide software development kits (SDKs) for all popular software languages. The SDKs replace the networking functions within the language. This means that the connections to and from the software are encrypted in the application space and are never exposed. In today's world of hybrid cloud and emerging architectures, such as mobile edge compute (MEC), this capability means you can secure your software, and the information it contains, regardless of where it is deployed – zero trust and network components follow the application – as they are embedded. Not only can you secure the application, but the application also has zero trust built into it, meaning that the application requires no additional coding investment to be zero trust secure, regardless of where it is placed/hosted.

Having embedded NetFoundry technology directly into your application, you can also leverage that connectivity for additional use cases. NetFoundry provides certificate verification and configured access authorization "out of the box". In some situations, additional checks may be required such as the direct control of the secured network connection with additional options including auditing connections and services at any required level of granularity.

The nature of authentication in NetFoundry networks makes repudiation extremely difficult; auditors or forensic analysts can be certain what specific user took what specific action(s). This can also serve as a deterrent and control function for insiders once they understand this type of audit trail information is inherent in a NetFoundry environment. The architecture also provides the most agility in deployment, allowing rapid changes to environments for business reasons, disaster recovery, or other drivers with very little consideration required for the security of the resources themselves.

In parts one and two of this series, we looked at what malicious actors do after they locate a potential target via scanning. Brute force, stolen credentials, or credential stuffing attacks are worthless against a NetFoundry ZTNA implementation. The authentication of nodes in NetFoundry is done via X.509 certificates, verified against a private certificate authority. The brute-forcing of such a certificate is effectively impossible. There is no entry point on the network that allows any traffic prior to the completion of the two-way authentication.

Therefore, no passwords can be sprayed or otherwise attempted. Since keys are never sent or stored other than on the nodes, there is no way an attacker could gather the certificate from credential dump or other sources. There is simply no way for attackers to gain a foothold remotely to begin an attack chain.

Achieving true zero trust as a networking goal is now required in today's information age architecture. As information resources become more widely deployed, as MEC and cloud infrastructures are rapidly deployed to the edge, and users become more mobile every day, legacy network architectures are now invalidated.

NetFoundry provides a range of options to deploy zero trust network architectures and provides the unique option of embedding zero trust into software – true zero trust. NetFoundry delivers gains in cost-effectiveness, agility, and the resulting ease of conducting business to make a significant return on investment. **To secure your networks and improve your business security posture, please contact us at NetFoundry and let us help you achieve your security and business goals.**

Learn more with these resources:

All Papers in this series are posted here:

https://netfoundry.io/resources/#whitepapers

Web: https://netfoundry.io

Blog: https://netfoundry.io/about/blog/

Connect on Social Media:

Twitter: @NetFoundry
LinkedIn: https://www.linkedin.com/company/17955047

About The Author

Mike Gorman, Head of Security and Compliance, NetFoundry

Mike is a seasoned Network and Security Engineer and applies his decades of experience as the Head of Security and Compliance for NetFoundry. Mike designs and manages NetFoundry's Security and Compliance program (including SOC-2 reporting) to ensure NetFoundry operations and products are secured properly, while helping to maintain an agile and highly automated development processes. In addition, Mike is a senior member of NetFoundry's engineering team, and in this role has developed monitoring and measuring systems, cloud deployment automation and DevOps tooling. Prior to NetFoundry Mike held a variety of Security and Solution Engineering roles with IBM, and Verizon.

About NetFoundry

NetFoundry is the leader in Cloud Native Networking, enabling businesses to simply, securely and cost effectively connect distributed applications across edges, clouds and service meshes. The NetFoundry platform, delivered as NaaS, enables businesses to connect applications without the costs and complexity of VPNs, custom hardware, and private circuits. NetFoundry's platform is accessed via APIs, SDKs, and DevOps tools integrations, enabling practitioners, application developers, and network administrators to get the levels of automation and agility which are only possible with connectivity-as-code.

*Disclaimer:
The information provided in this paper is offered “as-is” and is provided for general information purposes only. All information is provided in good faith. NetFoundry makes no representation or warranty of any kind, expressed, or implied in any way regarding the accuracy, adequacy, validity, availability or completeness of any information contained in the paper.