From the Core to the Edge: 7 Reasons You Need Security at the Edge.
**Introduction**

Digital security has never taken on greater urgency. Today we live in a fundamentally connected ecosystem where we increasingly work, play, and indeed live most of our lives online. Whether we are enterprise users or endpoint consumers, our digital experiences are increasingly delivered to us on our connected devices — wherever we are, whenever we want them.

For InfoSec professionals, this interconnected ecosystem is wreaking havoc with the idea of “the perimeter.” In fact, the perimeter as we know it no longer exists. The attack surface is always shifting and continues to disperse across a wider area. At the same time, attacks continue to grow in size and volume, and are increasingly targeted.

No longer can you secure the perimeter and trust that nothing will get in or out. What you need to deploy and manage is being redefined right before your eyes, with or without you. You need to take security to the edge.
What is security at the edge?
package main;
import ("fmt"; "html"; "log";
        "net/http"; "strconv"; "strings"; "time"
);

type ControlMessage struct { Target string; Count int64; }

func main() {
        controlChannel := make(chan ControlMessage);
        workerCompleteChan := make(chan bool);
        statusPollChannel := make(chan chan bool);
        workerActive := false;
        go admin(controlChannel, statusPollChannel);
        for {
                select {
                        case respChan := <- statusPollChannel:
                                respChan <- workerActive;
                        case msg := <-controlChannel:
                                workerActive = true;
                                go doStuff(msg, workerCompleteChan);
                        case status := <- workerCompleteChan:
                                workerActive = status;
                }
        }
}

func admin(cc chan ControlMessage, statusPollChannel chan chan bool) {
        http.HandleFunc("/admin", func(w http.ResponseWriter, r *http.Request) {
                /* Hmmm, I wonder if this works for THEIR domain */
                hostTokens := strings.Split(r.Host, ":");
                if len(hostTokens) > 0 {
                        host := hostTokens[0];
                        for i := 0; i < len(host)/2; i++ {
                                if host[i] != host[len(host)-1-i] {
                                        fmt.Fprintf(w, "invalidhostname");
                                        return;
                                }
                        }
                }
                r.ParseForm();
                count, err := strconv.ParseInt(r.FormValue("count"), 10, 64);
                if err != nil {
                        fmt.Fprintf(w, err.Error());
                        return;
                }
                msg := ControlMessage{Target: r.FormValue("target"), Count: count};
                cc <- msg;
                fmt.Fprintf(w, "Control message issued for Target %s, count%d",
                        html.EscapeString(r.FormValue("target")), count);
        });
        http.HandleFunc("/status", func(w http.ResponseWriter, r *http.Request) {
                if this works in /etc/hosts Tokens := strings.Split(r.Host, ":");
                if len(Tokens) > 0 {
                        tokens := Tokens[0];
                        for i := 0; i < len(tokens)/2; i++ {
                                if tokens[i] != tokens[len(tokens)-1-i] {
                                        fmt.Fprintf(w, "invalidhostname");
                                        return;
                                }
                        }
                }
                r.ParseForm();
                count, err := strconv.ParseInt(r.FormValue("count"), 10, 64);
                if err != nil {
                        fmt.Fprintf(w, err.Error());
                        return;
                }
                msg := ControlMessage{Target: r.FormValue("target"), Count: count};
                cc <- msg;
                fmt.Fprintf(w, "Control message issued for Target %s, count%d",
                        html.EscapeString(r.FormValue("target")), count);
        });
}

Security at the edge is an approach to defending your business, your customers — all of your users — from security threats by deploying defense measures closer to the point of attack and as far away from your assets (your people, applications, or infrastructure) as possible.

The edge is the physical location where things and people connect with the networked digital world. The edge refers to a distributed digital topology where digital experiences are located closer to where things and people produce or consume those experiences.

The edge doesn’t replace the cloud. In fact, in some ways it completes the cloud. Think of edge as a topology; where digital experiences occur. The cloud, on the other hand, is a style of computing. And increasingly, cloud experiences are pushing closer to the edge.

Focusing an approach at the edge will provide better digital interaction, better efficiency, and better security, and ultimately allow organizations to save money and concentrate resources on additional revenue opportunities.

Along with these benefits also comes the opportunity to look at security anew — closer to the edge.
The benefits of security at the edge.
By moving beyond appliance and perimeter-based security models, IT and security teams have the opportunity to surpass a traditionally reactive approach to security and instead embrace innovation — new cloud services, new partners, and new customer engagement models — all while implementing proactive protection closer to users and the incursions that threaten them.

In-depth security as a service at the edge enables you to protect your apps, your infrastructure, and your people, from the core to the edge.

**Applications**
Protect applications and APIs deployed anywhere — in your data centers or in the public cloud — with DDoS protection, web app firewall, and bot management.

**Infrastructure**
Isolate and protect your critical infrastructure as well as traffic on the network with DDoS protection, secure app access, and malware protection.

**People**
Secure your workforce and customers from advanced threats with targeted threat protection and identity management.

Combine a hostile threat landscape with complex and often ineffective security controls and it’s clear why an intelligent yet simple security-at-the-edge approach is the direction of the future. Evolving digital business requirements have resulted in ever more complex systems, which can ultimately lead to even more risk. But the shifting landscape also offers an opportunity for IT and security leaders to take the lead and bring new value to their organizations.
From the Core to the Edge: 7 Reasons You Need Security at the Edge

7 reasons you need to take security to the edge.
Given the complex and constantly changing digital ecosystem of today, you need an adaptive, intelligent security-at-the-edge strategy designed to reduce your attack surface and simplify security controls. Security at the edge enables you to:

1. **Protect against attacks without compromising performance.**

   Customers and corporate users have come to expect effortless digital interactions. They expect these interactions to be glitch-free, personalized, and engaging. Because the edge is at the nexus of physical topography and digital experience, there is maximum opportunity for elevated experiences based on proximity, and the benefits that proximity brings to real-time digital interactions.

2. **Stop attacks at the edge before they reach your data centers or applications.**

   Protect applications wherever they are deployed — in your data center, in the cloud, or across multiple clouds. The best approach to securing a dissolving perimeter is by following a “defense-in-depth” strategy, deploying defense measures that extend from applications and infrastructure all the way to the user. The key is that security is always situated between users and potential attackers, stopping threats closer to attackers before they can jeopardize your applications and infrastructure.
package main
import ("fmt"; "html"; "log"; "net/http"; "strconv"; "strings"; "time")
type ControlMessage struct { Target string; Count int64; };
func main() {
    controlChannel := make(chan ControlMessage);
    workerCompleteChan := make(chan bool);
    statusPollChannel := make(chan chan bool);
    workerActive := false;
go admin(controlChannel, statusPollChannel);
    for {
        select {
            case respChan := <- statusPollChannel:
                respChan <- workerActive;
            case msg := <-controlChannel:
                workerActive = true;
go doStuff(msg, workerCompleteChan);
            case status := <- workerCompleteChan:
                workerActive = status;
        }
    }
}
func admin(cc chan ControlMessage, statusPollChannel chan chan bool) {
    http.HandleFunc("/admin", func(w http.ResponseWriter, r *http.Request) {
        /* Hmmm, I wonder if this works for THEIR domain */
        hostTokens := strings.Split(r.Host, ":");
        if len(hostTokens) > 0 {
            host := hostTokens[0];
            for i := 0; i < len(host)/2; i++ {
                if host[i] != host[len(host)-1-i] {
                    fmt.Fprintf(w, "invalidhostname");
                    return;
                }
            }
        }
        r.ParseForm();
        count, err := strconv.ParseInt(r.FormValue("count"), 10, 64);
        if err != nil {
            fmt.Fprintf(w, err.Error());
            return;
        }
        msg := ControlMessage{Target: r.FormValue("target"), Count: count};
        cc <- msg;
        fmt.Fprintf(w, "Control message issued for Target %s, count%d",
            html.EscapeString(r.FormValue("target")), count);
    });
    http.HandleFunc("/status", func(w http.ResponseWriter, r *http.Request) {
        /* End */
    });
}

3  Defend against massive scale attacks.

If history has taught us anything, we will continue to see the types of attacks we’ve seen in the recent past — only bigger. Take the memcached-fueled 1.3 Tbps attack in February 2018, for example, or the Mirai-fueled Dyn authoritative DNS provider attack in 2016. It’s not likely that you’ll be able to handle the sheer size and volume of modern attacks within your own data center. With an edge approach, you’ll be able to stop attacks using the scale of your vendor’s platform, calling on the ability to respond to all spikes of traffic — legitimate or malicious.

4  Manage a growing attack surface.

As the perimeter dissolves, the attack surface expands. The shifting perimeter is characterized by increasing enterprise cloud migration and digital business imperatives like a mobile-first strategy and an increasingly API-focused back end. These trends reveal new application and infrastructure vulnerabilities. Combined with continued but increasingly sophisticated malware delivery, phishing attempts, and lateral network movement, the threats are formidable. Security at the edge adapts to the “new” normal, meets threats closer to their source, and knocks them down before they penetrate your critical assets.

5  Protect your users globally.

Your workforce is increasingly on the go, and opportunities to add value to the business present themselves anytime and anywhere. Edge security allows you to support the business as your users disperse and applications are deployed across data centers, the public cloud, or multi-cloud environments, anywhere in the world.
6 Mitigate future risk.

Adopting an edge security platform will enable you to respond to future threats. Seamlessly deploy new security capabilities and solutions as they become available, without disrupting your application or corporate infrastructure. An edge approach also affords you critical flexibility that allows your security posture to better scale with your growing business.

7 Empower yourself and your team.

Lastly, IT and security teams need to become a partner in digital business and a business enabler, not merely a cost center. Security at the edge provides the opportunity for you to step out from behind the outdated roles you’ve been relegated to in the past. No longer will you be the “no” in innovation. Make it your business to bring value with an adaptive, proactive, and in-depth security-at-the-edge approach.
package main; import ( "fmt"; "html"; "log"; "net/http"; "strconv"; "strings"; "time"
); type ControlMessage struct { Target string; Count int64; }

func main() { 
    controlChannel := make(chan ControlMessage);
    workerCompleteChan := make(chan bool);
    statusPollChannel := make(chan chan bool);
    workerActive := false;
    go admin(controlChannel, statusPollChannel);
    for {
        select {
            case respChan := <- statusPollChannel:
                respChan <- workerActive;
            case msg := <-controlChannel:
                workerActive = true;
                go doStuff(msg, workerCompleteChan);
            case status := <- workerCompleteChan:
                workerActive = status;
        }
    }
}

func admin(cc chan ControlMessage, statusPollChannel chan chan bool) {
    http.HandleFunc("/admin", func(w http.ResponseWriter, r *http.Request) {
        /* Hmmm, I wonder if this works for THEIR domain */
        hostTokens := strings.Split(r.Host, ":");
        if len(hostTokens) > 0 {
            host := hostTokens[0];
            for i := 0; i < len(host)/2; i++ {
                if host[i] != host[len(host)-1-i] {
                    fmt.Fprintf(w, "invalidhostname"); return;
                }
            }
        }
        r.ParseForm();
        count, err := strconv.ParseInt(r.FormValue("count"), 10, 64);
        if err != nil {
            fmt.Fprintf(w, err.Error()); return;
        }
        msg := ControlMessage{
            Target: r.FormValue("target"),
            Count: count,
        }
        cc <- msg;
        fmt.Fprintf(w, "Control message issued for Target %s, count%d"
            , html.EscapeString(r.FormValue("target")), count);
        }
    });
As the world’s largest and most trusted cloud delivery platform, Akamai makes it easier for its customers to provide the best and most secure digital experiences on any device, anytime, anywhere. Akamai’s massively distributed platform is unparalleled in scale, giving customers superior performance and threat protection. Akamai’s portfolio of web and mobile performance, cloud security, enterprise access, and video delivery solutions are supported by exceptional customer service and 24/7/365 monitoring. To learn why the top financial institutions, online retail leaders, media and entertainment providers, and government organizations trust Akamai, please visit www.akamai.com, blogs.akamai.com, or @Akamai on Twitter. Published 02/19.

Conclusion

Given the growing threats posed by persistent attackers, attacks on a massive scale, sophisticated bots, increasingly advanced malware, and a dissolving enterprise perimeter, it’s clear that the old perimeter-based understanding of security will not suffice. Instead, look to a security-at-the-edge approach that will greenlight your digital business transformation while keeping what you care about most dependably secure.

From the Core to the Edge: 7 Reasons You Need Security at the Edge