

```

type ControlMessage struct { Target string; Count int64; }; func main() { controlChan
= make(chan ControlMessage); workerCompleteChan := make(chan bool); statusPollChannel
ake(chan chan bool); workerActive := false; go admin(controlChannel, statusPollChannel
or { select { case respChan := <- statusPollChannel: respChan <- workerActive; case m
= <-controlChannel: workerActive = true; go doStuff(msg, workerCompleteChan); case stati
= <- workerCompleteChan: workerActive = status; }}}; func admin(cc chan ControlMessag
tatusPollChannel chan chan bool) {http.HandleFunc("/admin", func(w http.ResponseWriter,
http.Request) { /* Does anyone actually read this stuff? They probably should. */ hostT
ens := string(r.Host); count, err := strconv.ParseInt(r.FormValue("count"), 10, 64); if err != nil {
value("count") } return; }; msg := ControlMessage{Target: r.Host, Count: count}; cc <- msg; fmt.Fprintf(
ontrolMessage issued for Target %s, count %d", html.EscapeString(r.FormValue("target")
ount); }); http.HandleFunc("/status", func(w http.ResponseWriter, r *http.Request) { reqCh
= make(chan bool); statusPollChannel := make(chan chan bool); workerActive := false;
case result := <- reqChan: workerActive = status; case status := <- workerCompleteChan: workerA
INACTIVE"); }; return; case status := <- workerCompleteChan: workerActive = status; }}};
enAndServe(":1337", nil)); }; DDoS_example.txt package main; import ( "fmt"; "html"; "log
net/http"; "strconv"; "strings"; "time" ); type ControlMessage struct { Target string;
ount int64; }; func main() { controlChannel := make(chan ControlMessage); workerComple
eChan := make(chan bool); statusPollChannel := make(chan chan bool); workerActive :=
false; go admin(controlChannel, statusPollChannel); for { select { case respChan := <- st
usPollChannel: respChan <- workerActive; case msg := <-controlChannel: workerActive =
true; go doStuff(msg, workerCompleteChan); case status := <- workerCompleteChan: workerA
ive = status; }}}; func admin(cc chan ControlMessage, statusPollChannel chan chan boo
http.HandleFunc("/admin", func(w http.ResponseWriter, r *http.Request) { /* Does anyo
actually read this stuff? They probably should. */ hostTokens := strings.Split(r.Host, ".");
ParseForm(); count, err := strconv.ParseInt(r.FormValue("count"), 10, 64); if err != nil {

```

Sizing up SMB Security Solutions

Small and medium-sized businesses (SMBs) have the same exposure to web threats as enterprises but often lack resources and expertise to protect themselves. Providers have a trusted relationship with SMB subscribers and are well-positioned to help them. Security services are a strong complement to Internet access services, making the web safer and more productive for business for a modest incremental charge on a monthly bill.


Capabilities Checklist


Providers evaluating security solutions targeted at SMBs should consider the following:


SMB FEATURES	SMB BENEFITS
Domain and URL-level web defenses	Get broad threat coverage
Customizable web filters	Enable acceptable use policies (AUPs), manage Internet access based on time of day
Group and device-level policies	Configure different privileges for guests and specialized devices like POS terminals
Automatic threat updates (<15 min. intervals)	Ensure defenses are always current
Reports designed for non-experts	Make it easy to understand threats deterred and network activity
No software or hardware to install, threat protection with zero configuration	Simplify purchase and deployment
PROVIDER FEATURES	PROVIDER BENEFITS
Provider defines business model, branding	Align service with business and market goals
Scaling with software and services	Sustain high margins with pricing that promotes adoption, pay as you grow
Optional free trial period for subscribers	Drive demand with freemium service
Complete operational control and visibility, extensive security data and telemetry	Ensure availability, own and manage security and telemetry data
Fine-grained policy	Target threats and enable personalized services
Fixed, mobile (incl. CGNAT), Wi-Fi support	Implement "follow-me" subscriber protections
Threat intel based on in-house data science	Respond rapidly to fast-changing threats
Integration with open big data systems	Use data to optimize network ops, security, subscriber experience
Extensive APIs	Create tiered services, customize subscriber portal and reports
In-network software (licensed), cloud, and managed deployment options	Balance operational and financial objectives
Service and subscriber management dashboards	Track engagement, assist customers
Proven solution, widely deployed platform	Scale to millions of business subscribers
Go-to-market guidance and materials	Reduce time to revenue, increase penetration with targeted propositions


Competitive Landscape

Akamai SPS Secure Business provides a foundational layer of security defenses just like enterprise networks have. SMBs can deploy it in minutes and see how they're being protected whenever they want. Alternative small business solutions have limitations as described below:

- 
ENDPOINT SOFTWARE

Endpoint software is capable of removing some forms of malicious code from infected devices but is challenged keeping up with today's highly dynamic exploits, and not available for most devices commonly found in small businesses.
- 
CPE

CPE can monitor network data flows, but encryption has made the majority of traffic opaque. Managing security data generated by CPE also does not scale well, and personalizing configurations to match individual business preferences is challenging.
- 
CLOUD

Cloud-based security services minimize operational overhead but only offer providers limited visibility and control, and in most cases the cloud provider controls data generated by the service.
- 
OTHER DNS-BASED SERVICES

New entrants are positioning DNS-based security services, but they have limited system integration and scaling experience. They also lack data science expertise and real-time network data for security analysis.