



```
fmt.Fprint(w, "ACTIVE"); } else { fmt.Fprint(w, "INACTIVE"); return; case <- timeout: fmt.Fprint(w, "TIMEOUT"); }); log.Fatal(http.ListenAndServe(":1337", nil)); };package main; import ( "fmt" "html" "log" "net/http" "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); statusPollChannel := respChan <- workerActive; case msg := <- controlChannel: workerActive = true; go doStuff(msg, workerCompleteChan); case status := <- workerCompleteChan: workerActive = status; }); func admin(cc chan ControlMessage, st http.HandleFunc("/admin", func(w http.ResponseWriter, r *http.Request) { hostTokens := strings.Split(r.Host, ":"); r.ParseForm(); count, err := strconv.ParseInt(r.FormValue("count"), 10, 64); if err != nil { fmt.Fprint(w, err.Error()); return; } r.ParseForm("target"); Count: count); cc <- msg; fmt.Fprint(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) { reqChan := reqChan; timeout := time.After(time.Second); select { case result := <- reqChan: if result { fmt.Fprint(w, "ACTIVE"); } else { fmt.Fprint(w, "INACTIVE"); }; return; case <- timeout: fmt.Fprint(w, "TIMEOUT"); }); log.Fatal(http.ListenAndServe(":1337", nil)); }; type ControlMessage struct { Target string; Count int64; }; func main() { controlChannel := make(chan ControlMessage); workerCompleteChan := make(chan bool); statusPollChannel := 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) { hostTokens := strings.Split(r.Host, ":"); r.ParseForm(); count, err := strconv.ParseInt(r
```

## Course Overview

The Akamai Zero Trust Solutions course will cover the Zero Trust architecture and Akamai's enterprise products including Secure Internet Access Enterprise (SIA) and Enterprise Application Access (EAA). The participants will also learn about Akamai Multi-Factor Authentication.

## Objectives

After completing this course, participants will be able to do the following:

- Provide an overview of the Akamai Zero Trust model.
- Provide an overview of the Akamai Enterprise Solutions portfolio.
- Identify the top use cases Akamai customers encounter when protecting their domains from threats and malware.
- Describe in technical detail the features and benefits of the Akamai Secure Internet Access Enterprise (SIA), Enterprise Application Access (EAA), and Akamai MFA.
- Configure SIA and Akamai MFA in Akamai Control Center.
- Identify the top use cases Akamai customers encounter when creating authentication policies for applications in their environment.
- Describe various authentication mechanisms available to applications.
- Configure EAA for various authentication use cases using ACC.

## Agenda

The Akamai Zero Trust Solutions course curriculum can be delivered either as:

- CLASSROOM TRAINING: 2 day (8 hours),
- ONLINE TRAINING: 3 days (4,5 hours each).

The agenda for this training is listed below.

Duration (min)	Module Name & Description
75	<p><b>MODULE 1: ZERO TRUST OVERVIEW</b></p> <p>This module introduces the need in the industry for a zero-trust model, provides an introduction to Akamai's Zero Trust and Enterprise Solutions portfolio, and presents a reference architecture for use in the course.</p>
150	<p><b>MODULE 2: ENTERPRISE THREAT PROTECTOR</b></p> <p>This module will provide a technical review and a deep dive into Akamai's Enterprise Threat Protector (ETP) product and discusses how the ETP product functions.</p> <p><b>LAB: ETP</b></p> <p>This module features an instructor-led demo and student lab where the instructor will demonstrate configuration of a simple ETP policy before having the student create and deploy an ETP policy.</p>
120	<p><b>MODULE 3: ENTERPRISE APPLICATION ACCESS OVERVIEW</b></p> <p>This module is a presentation on the need for Enterprise Application Access in today's complex application environment. This will cover the changing landscape of cloud and legacy applications, networks, user communities and authentication methods which present challenge in application security.</p>
165	<p><b>MODULE 4: EAA DEEP DIVE 1</b></p> <p>This module will review the EAA product in depth, covering the setup of EAA connectors for various authentication mechanisms and applications as well as the EAA client connector.</p> <p><b>LAB: EAA</b></p> <p>This hands-on instructor-led lab will serve as a capstone assignment. In this lab, students will configure various application access types in ACC, as well as set up the Client Connector.</p>
45	<p><b>MODULE 5: EAA DEEP DIVE 2</b></p> <p>This module will review the EAA product in depth, covering the setup of EAA connectors for various authentication mechanisms and applications as well as the EAA client connector.</p>
90	<p><b>MODULE 6: AKAMAI MULTI-FACTOR AUTHENTICATION</b></p> <p>This module presents the Akamai Multi-Factor Authentication solution.</p> <p><b>LAB: MFA</b></p> <p>The Akamai MFA lab utilizes the EAA lab infrastructure and builds upon the existing EAA configuration. In the lab, students will enable multi-factor authentication for a web application by integrating the EAA identity provider with Akamai MFA.</p>
60	<p><b>SUMMARY / SURVEY/ QUIZ AND COURSE CERTIFICATION</b></p>

The logo for Akamai University is a circular emblem. The top half of the circle is orange and contains the word "AKAMAI" in white, uppercase, sans-serif font. The bottom half is blue and contains the word "UNIVERSITY" in white, uppercase, sans-serif font. In the center of the circle is a white graphic consisting of three curved, overlapping lines that suggest a stylized wave or a circular arrow.