

OPSTEC ON THE HIGH SEAS: A GOPHISH ADVENTURE



ABOUT CW LABS :

CW Labs is a renowned UK based Ed-tech company specializing in cybersecurity cyber range labs. They provide on-demand educational services and recognize the need for continuous adaptation to evolving threats and client requirements. The company has two primary divisions:

1. Cyber Range Labs
2. Up-Skilling Platform



ABOUT SPEAKER :

ABHIJEET KUMAR {SECURITY RESEARCHER}

His areas of interests includes Red Team Operations, Network Security, Cloud Infrastructure, and Linux Systems. Apart from this, he enjoys researching Adversarial TTPs and experimenting in his homelab.



DISCLAIMER

- ☺ *The information provided in this webinar is for educational purposes only*
- ☺ *We (**Organizer and Presenter**) do not endorse or support any illegal or unethical actions*
- ☺ *Attendees are solely responsible for how they use the knowledge gained from this webinar*



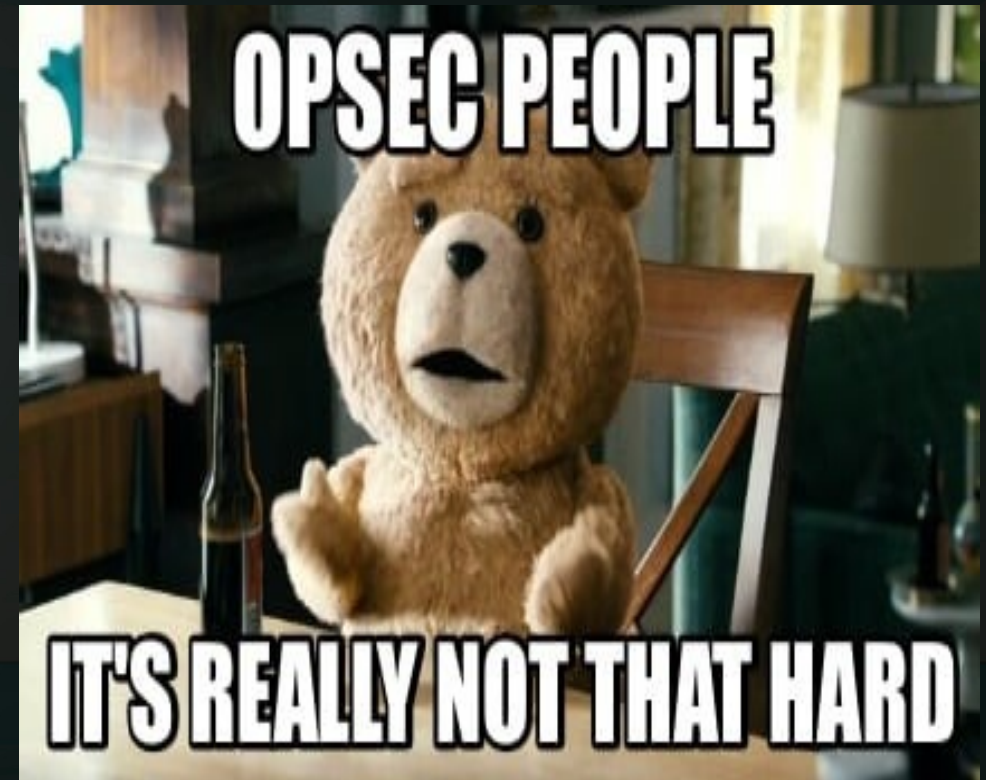
GOPHISH 101

- 😊 Phishing simulation tool
- 😊 Open source
- 😊 Written in GO
- 😊 Easy to deploy
- 😊 Highly customizable



OPSEC 101 FOR GOPHISH

- ☺ Stands for Operational Security
- ☺ Protect internal operations
- ☺ Prevent disclosure of sensitive information
- ☺ Remove signed artifacts



ARTIFACTS OF INTEREST

- ☺ Server Name
- ☺ Port(s)
- ☺ Testing Email Message
- ☺ Default Headers
- ☺ RID Parameter
- ☺ Default TLS Certificate
- ☺ 404 Not Found



SERVER NAME

- ☺ Identifier of the Gophish server
- ☺ Configuration at :

👉 `config > config.go`

```
44 // ServerName is the server type that is returned
45
46 const ServerName = "gophish"
47
48 //
49 fu
50
51
52 >
53
54
55
56 >
57
58
59
60
```

Constant ServerName in github.com/gophish/gophish/config/config.go 8 usages

File	Line	Usage
config.go config	45	// ServerName is the server type that is returned
phish.go controllers	212	w.Header().Set("X-Server", config.ServerName) //
phish.go controllers	306	Server: config.ServerName,
phish_test.go controllers	148	Server: config.ServerName,
email_request.go models	121	msg.SetHeader("X-Mailer", config.ServerName)
email_request_test.go models	81	"X-Mailer": config.ServerName,
maillog.go models	200	msg.SetHeader("X-Mailer", config.ServerName)
maillog_test.go models	269	"X-Mailer": config.ServerName,

PORT(S)

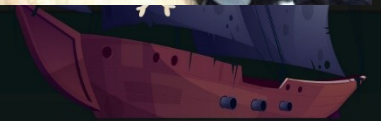
- ☺ Admin portal listens on port **3333**
- ☺ Configuration(s) at :
 - ↪ `config.json`
 - ↪ `config > config_test.go`
- ☺ Use SSH port forward :

```
ssh -L [local_addr]:[local_port]:[remote_addr]:[remote_port] [user]@[remote_ip] -i <keyfile.pem>
```



TESTING EMAIL MESSAGE

- ☹ Used for testing setup
- ☹ Huge red flags
- ☹ Alerts email providers
- ☹ Burns phishing domain(s)



CONTD...

☹ Configuration at :

➡ *Controllers > api > util.go*

```
// If a Template is not specified use a default
if s.Template.Name == "" {
    //default message body
    text := "It works!\n\nThis is an email letting you know that your gophish\nconfiguration was successful.\n"
        "Here are the details:\n\nWho you sent from: {{.From}}\n\nWho you sent to: \n" +
        "{{if .FirstName}} First Name: {{.FirstName}}\n{{end}}" +
        "{{if .LastName}} Last Name: {{.LastName}}\n{{end}}" +
        "{{if .Position}} Position: {{.Position}}\n{{end}}" +
        "\nNow go send some phish!"
    t := models.Template{
        Subject: "Default Email from Gophish",
        Text:    text,
    }
    s.Template = t
}
```



DEFAULT HEADERS

- ☺ Used for identification
- ☺ To communicate with web servers
- ☺ To filter out script kiddies
- ☺ To prevent abuse of Gophish



CONTD...

Configuration(s) at :

models > testdata > email_request.go

models > testdata >
email_request_test.go

models > testdata > maillog.go

models > testdata > maillog_test.go

models > testdata > smtp_test.go



RID PATAMETER

- Used for tracking campaigns
- Configuration at :

`model > testdata > campaign.go`

```
// RecipientParameter is the URL parameter that points to  
const RecipientParameter = "rid"
```

Constant RecipientParameter in github.com/gophish/gophish/models/camp: 13 usages

File	Line	Code Snippet
phish.go controllers	321	rid := r.Form.Get(models.RecipientParameter)
phish_test.go controllers	46	resp, err := http.Get(fmt.Sprintf("%s/track?%s=%s", ...))
phish_test.go controllers	65	resp, err := http.Get(fmt.Sprintf("%s/track?%s=%s", ...))
phish_test.go controllers	78	resp, err := http.Get(fmt.Sprintf("%s/report?%s=%s", ...))
phish_test.go controllers	90	resp, err := http.Get(fmt.Sprintf("%s/report?%s=%s", ...))
phish_test.go controllers	102	resp, err := http.Get(fmt.Sprintf("%s/?%s=%s", ...))
phish_test.go controllers	117	resp, err := http.Get(fmt.Sprintf("%s/?%s=%s", ...))
phish_test.go controllers	130	resp, err := http.Get(fmt.Sprintf("%s?%s=%s", ...))
phish_test.go controllers	398	client.PostForm(fmt.Sprintf("%s/?%s=%s", ...))
campaign.go models	129	// RecipientParameter is the URL parameter that p
email_request_test.go models	167	expectedURL := fmt.Sprintf("http://127.0.0.1/%s/?")
maillog_test.go models	336	expectedURL := fmt.Sprintf("http://127.0.0.1/%s/?")
template_context.go models	57	q.Set(RecipientParameter, rid)

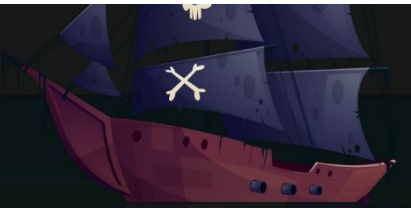


DEFAULT TLS CERTIFICATE

- ☺ Value is *Gophish*
- ☺ In default TLS certificate
- ☺ Configuration at :
 - ↪ *util > util.go*

Certificate

Gophish	
<hr/>	
Subject Name	
Organization	Gophish
<hr/>	
Issuer Name	
Organization	Gophish
<hr/>	
Validity	



CONTD...

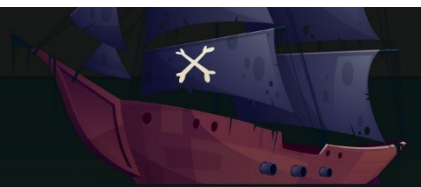
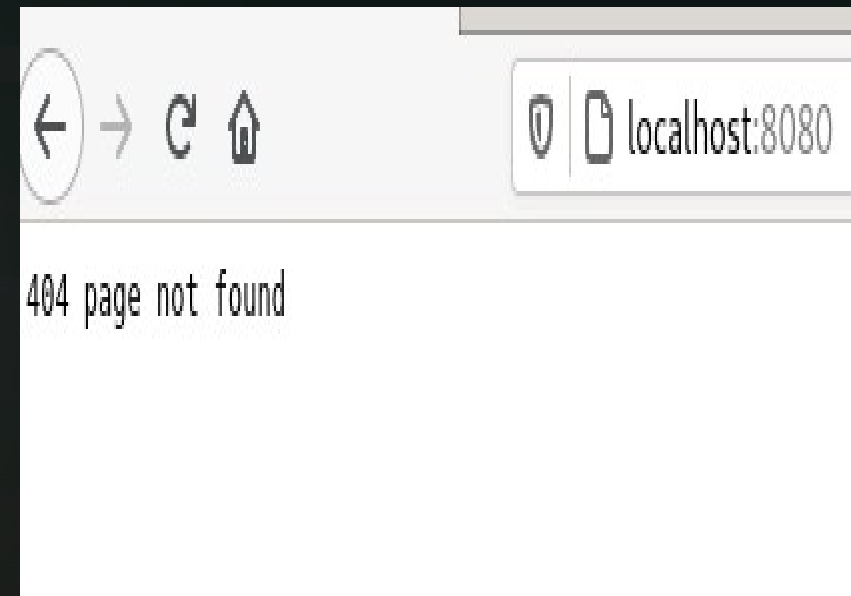
- ☺ Change to another string
- OR,
- ☺ Generate using *Let's Encrypt*

```
template := x509.Certificate{
    SerialNumber: serialNumber,
    Subject: pkix.Name{
        Organization: []string{"Gophish"},
    },
    NotBefore: notBefore,
    NotAfter: notAfter,
```



404 NOT FOUND

- ☹ On invalid URL requests
- ☹ Unique to Gophish
- ☹ Easy to detect
- ☹ Change to custom error page



HOW TO COMPILE?

- ☺ Open a terminal
- ☺ Install *GO* and *GCC*
- ☺ Navigate to Gophish directory
- ☺ Run “*go build*”
- ☺ Copy compiled binary to remote server using *SCP*



REFERENCES

- ① <https://github.com/gophish/gophish>
- ① <https://www.sprocketsecurity.com/resources/never-had-a-bad-day-phishing-how-to-set-up-gophish-to-evade-security-controls>
- ① https://github.com/puzzlepeaches/sneaky_gophish
- ① <https://blog.cybercx.co.nz/identifying-gophish-servers>





THANK YOU !

If you like the content, please feel free to shout out & tag us at social media platforms.

For any technical questions / doubts related to the content please email us at

support@cyberwarfare.live

For Professional Red / Purple Team Labs & Technical Training Services kindly email at

info@cyberwarfare.live

Cyberwarfare.live

