



OPSEC ON THE HIGH SEAS: A GOPHISH ADVENTURE



DISCLAIMER

- ☺ *The information provided in this talk 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 talk*



WHAMI

- 🕸 Security Researcher
- 🕸 Current gig :-
 - 👉 Doing offensive stuff @ CyberWarFare Labs
- 🕸 Breaker of Technology
- 🕸 Maker of food

AGENDA

- ☺ Gophish 101
- ☺ OpSec 101
- ☺ Artifacts of Interests
- ☺ How to Compile
- ☺ A step beyond

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 Indicator of Compromise (IoC) 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

👉 Used in Email Headers

☹ Configuration at :

👉 *config > config.go*

```
44
45 // ServerName is the server type that is returned
46 const ServerName = "gophish"
47
48 //
49 func
50
51
52 >
55
56
57 >
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 if no template is provided
- ☹ 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
- ☺ Headers include :-
 - ☞ X-Mailer
 - ☞ X-Gophish-Contact



CONTD...

④ Configuration(s) at :

➔ `models > email_request.go`

➔ `models > email_request_test.go`

➔ `models > maillog.go`

➔ `models > maillog_test.go`

➔ `models > smtp_test.go`



RID PARAMETER

- ☹ Used for tracking campaigns
- ☹ Configuration at :

➡ *models > campaign.go*

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

Constant RecipientParameter in github.com/gophish/gophish/models/campaign.go 13 usages

File	Line	Code
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", ct
phish_test.go controllers	117	resp, err := http.Get(fmt.Sprintf("%s/?%s=%s", ct
phish_test.go controllers	130	resp, err := http.Get(fmt.Sprintf("%s%s?%s=%s", i
phish_test.go controllers	398	resp, err := client.PostForm(fmt.Sprintf("%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

- ☺ TLS certificate used for Admin interface (*Port 3333*)
- ☺ Value is **Gophish** in default TLS certificate
- ☺ Configuration at :

➡ **util > util.go**

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


CONTD...

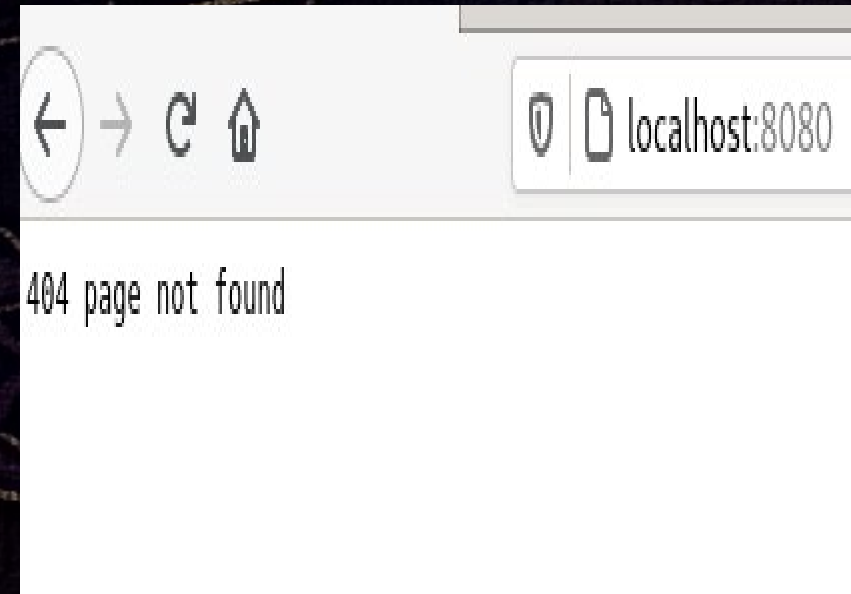
- ☹️ Change to another string
- OR,
- ☹️ Generate using **Let's Encrypt** (for free)

Certificate

<input type="text" value="Gophish"/>	
Subject Name	
Organization	<input type="text" value="Gophish"/>
Issuer Name	
Organization	<input type="text" value="Gophish"/>
Validity	

404 NOT FOUND

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



HOW TO COMPILE?

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

A STEP BEYOND

- ☺ Use domain reputation
 - 👉 Purchase expired domains
- ☺ Don't use own SMTP servers
 - 👉 Use providers like AWS SES, Mailgun, Sendgrid, etc
- ☺ Don't reuse infrastructure across engagements
 - 👉 Use IaC tools like Terraform + Ansible



REFERENCES

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

A dark, textured background featuring a pirate ship on the right side, a full moon in the upper center, and silhouettes of palm trees and a person on the left. The ship has a skull and crossbones on its sail.

THANK YOU

😊 Slides will be available at:

<https://github.com/wand3rlust/My-Presentations>