CWL
CyberWarFare Labs

OPSEC 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

CWL
CyberWarFare Labs

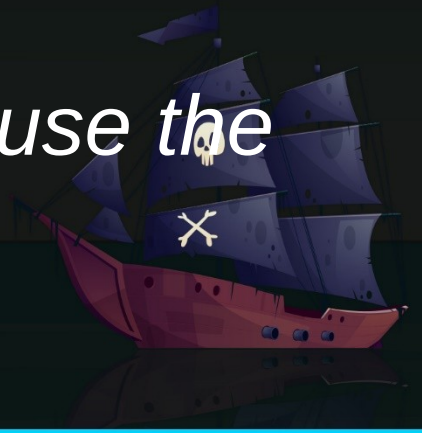
ABOUT SPEAKER :

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*

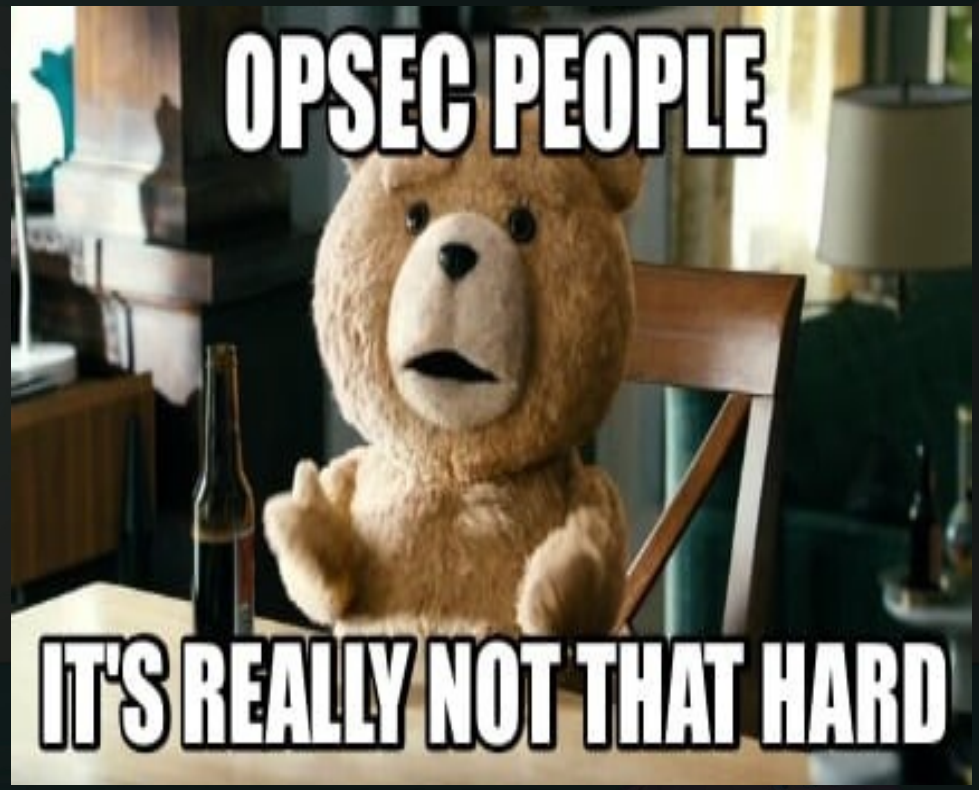


GOPHISH 101

- 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 signatured artifacts

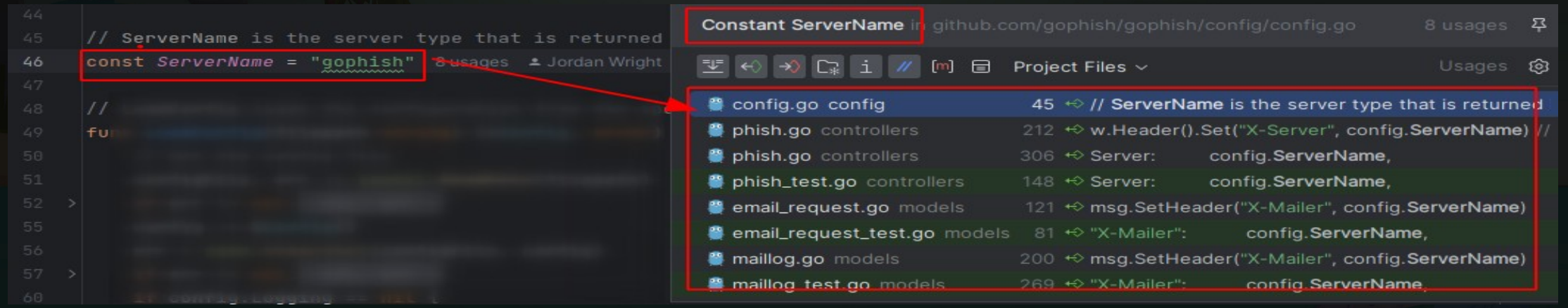


Artifacts of Interest

- 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*



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)

CWL
CyberWarFare Labs

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

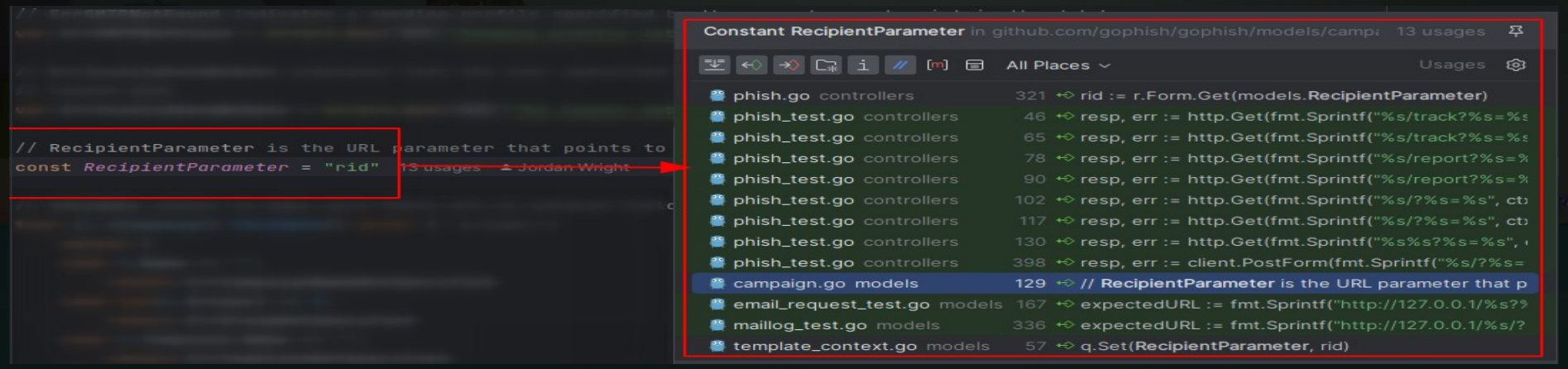
- Used for identification
- To communicate with web servers
- To filter out script kiddies
- To prevent abuse of Gophish

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*

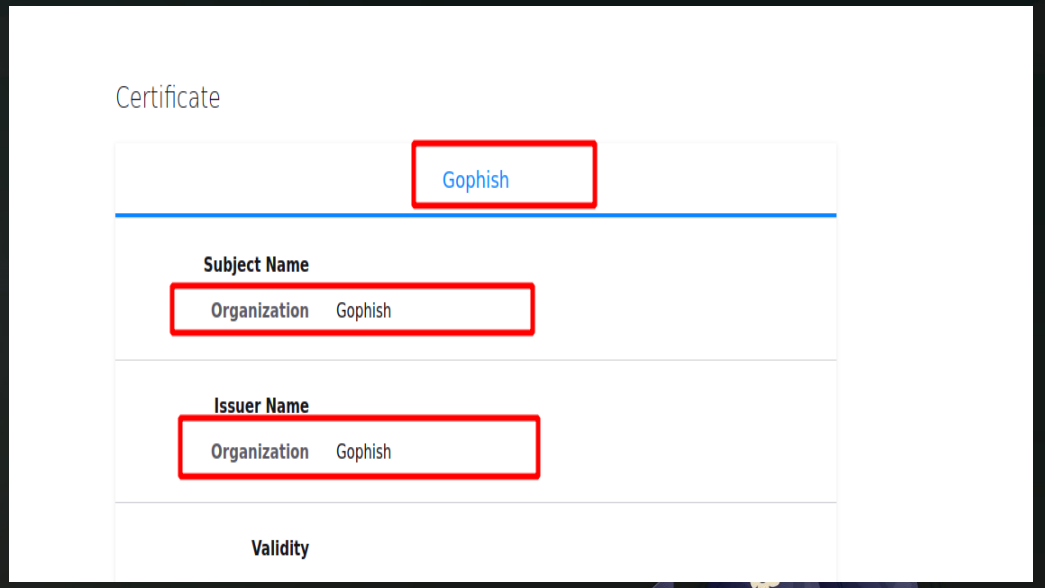
RID PATAMETER

- Used for tracking campaigns
- Configuration at :
  - *model > testdata > campaign.go*



DEFAULT TLS CERTIFICATE

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



CONTD...

- Change to another string

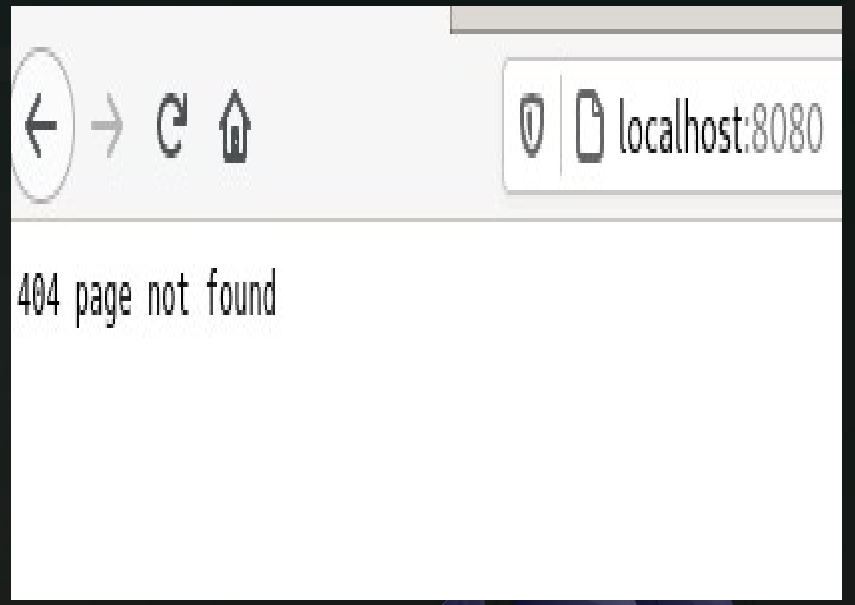
  OR,

- 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*

CWL
CyberWarFare Labs

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*

CWL
CyberWarFare Labs

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