PROFESIONALIZANDO NUESTRO PHISHING

Marta Barrio Marcos

Applications Security Architect at Beam Suntory

> 9 years of experience

CISA, CEH, CSX, OSCP, OSCE

Speaker at security conferences like NN2019, C1b3rWall Academy

Trainer: ISACA, HackBySecurity, master UCLM, EIP y UCAM

https://es.linkedin.com/in/martabarriomarcos
@martrudix - @securiters

Carolina Gómez Uriarte

Pentester at VASS

> 3 years of experience

Head of Sh3llCON

CEH

Speaker at security conferences like NN2019, HoneyCON, C1b3rWall Academy

Author at sh3llcon.org

https://es.linkedin.com/in/carolina-gomez-uriarte
@Carol12Gory

#index

1.
Part I

Introduction

Disclaimer

This session is held to raise awareness and teach how cyber criminals carry out phishing attacks.

Everything explained in this session has been done in controlled environments and without any evil purpose.

Both speakers are not responsible for any illegitimate use for profit.

What is phishing?

- Impersonation of web pages, e-mails, etc.
- Use of logos, texts, images known by the user to mislead the user and make him fall for the trick.
- Target: credentials, malware distribution, subscriptions to spam lists...

How to detect phishing?

- Check the sender
- Check the grammar of the mail
- Where does the link go?
- Others checks like Mail Headers: IPs, domains, etc.

Phishing examples

De: Liberbank <clienteee@liberbank.es>
Fecha: 3 jun. 2019 8:34 p. m.
Asunto: Notificación
Para:
Cc:

Liber bank

Buenos dias,

Para evitar el uso fraudulento de tarjetas de crédito en Internet, **liberbank** tiene un nuevo sistema de control de pagos.

Este servicio es completamente gratis.

Nuestro sistema ha detectado que no activa su servicio de **Clave OTP**

Para activar este servicio, simplemente haga clic en el siguiente enlace y siga los pasos provistos :

Acceso clientes

Saludos ,
Carmen Maria Marchal Basalo.

Este email es resultado de una investigacion **liberbank** S.A.

http://nuevadigital.co.vu/?ref=9809C51RO2M5BB3908H8SY1HXTFRLW3D0998D7H897

Phishing examples

Correos
El equipo de correos: Su paquete est·esperando la entrega.
Para: Instituto Superior de Ciberseguridad

Entrada - Iscibersegurid 3:06

Correos

Estimado cliente,

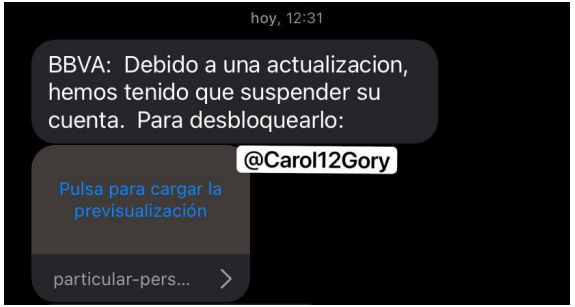
Verifica tu tarjeta de crédito y su paquete est·esperando la entrega. Confirme el pago en el siguiente enlace, la verificaciôn en lÌnea debe hacerse en los prôximos 15 dÌas antes de que caduque, Siga las instrucciones :

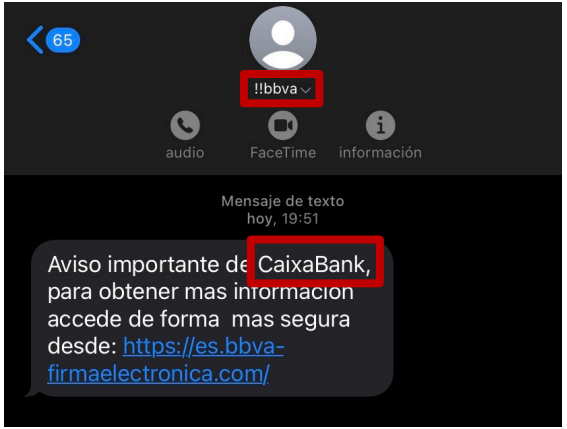
Haga clic aqui:

saludos,
El equipo de correos,

https://twitter.com/juliocesarlopd/status/1331142405052100609?s=20

Phishing examples

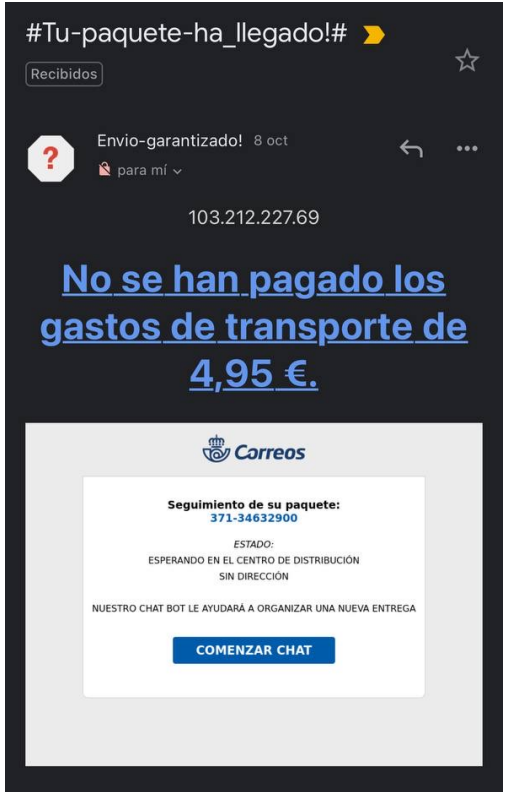




https://twitter.com/carol12gory/status/1416094066828267521?s=28

Phishing examples

Phishing examples



De: **C0NFirmación** <makeit.fxiltybygjydenpfipmyzr@hipwireless.org.uk>
Date: mar., 5 oct. 2021 12:37
Subject: C0NFirma..TuPaquete..
To: <crashbboy1@gmail.com>
Cc: <crashbboy1@gmail.com>

PaQueTE:3*1-*******00

Correos

Seguimiento de su paquete:
371-34632900

ESTADO:
ESPERANDO EN EL CENTRO DE DISTRIBUCIÓN
SIN DIRECCIÓN

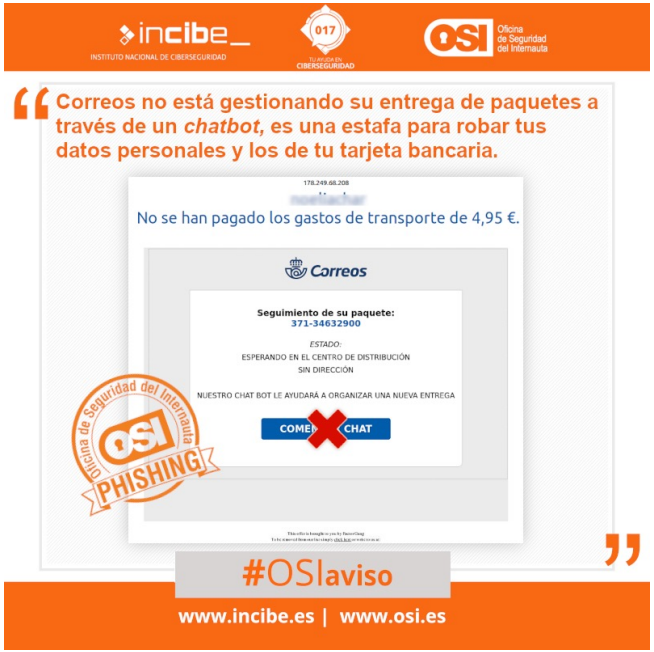
NUESTRO CHAT BOT LE AYUDARÁ A ORGANIZAR UNA NUEVA ENTREGA

COMENZAR CHAT

This offer is brought to you by BetterForm
To be removed from our list simply click here or write to us at:
BetterForm 4050 Pennsylvania Ave.#90 Kansas City MO 64111-3041

hipwireless.org.uk/rc0a5.php?32=1o09615c174dd3bb3_0u34.4wcrduy.A00vyrfnyqs1q517ti_nq1083.00r6lMGJmZWUwYmx1a2lq0m4NPa

Phishing examples



https://twitter.com/osiseguridad/status/1448650526447464452?s=20

2.
Part II

Hands on labs

Requirements
Deployment

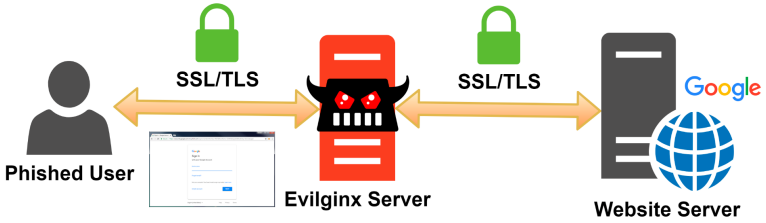
- **GoPhish** – Mail sender
- **Evilginx** – Manage the phishing

GoPhish
Deployment

- *Config.json* file → IP and certificates config
- Sending Profiles → Config SMTP relay.
- ~~Landing Pages → Web that is shown to the victim when he clicks on the link~~
- Email Templates →Mail received by the victim
- Users & Groups → Destinations
- Campaigns

Evilginx2
Deployment

- **Phishlet** → YAML file where are defined subdomain necessary to do proxy, which strings should be replaced in relayed packets and which cookies should be captured.
- **Lure** → generation of phishing URLs
- **Config** → general configuration
- **Sessions** → sessions and captured tokens with credentials



@mrgretzky


ENOUGH TALK
TIME TO DEMO

3.
Part III

How not to be phished

Countermeasures
Recommendations for final users

- Check domain in URL bar
- Use U2F devices
- DO NOT use SMS 2FA – SIMJacking
- Common sense

Countermeasures
Recommendations for developers

- Check *window.location*
- Check *window.location* & obfuscate

```
» window.location
← ▾ Location https://www.google.com/
    ▸ assign: function assign()
      hash: ""
      host: "www.google.com"
      hostname: "www.google.com"
      href: "https://www.google.com/"
      origin: "https://www.google.com"
      pathname: "/"
      port: ""
      protocol: "https:"
    ▸ reload: function reload()
    ▸ replace: function replace()
      search: ""
    ▸ toString: function toString()
    ▸ valueOf: function valueOf()
      Symbol(Symbol.toPrimitive): undefined
```

Countermeasures
Recommendations for developers

- Check *baseURI* property of DOM items

```
>> $("body").baseURI
<- "https://www.google.com/"
```

- Check headers: *X-Evilginx* or *X-Mailer*

```
Date: Thu, 22 Apr 2021 10:44:29 +0200
From: Outlook <security@microsoft-outlook.com>
X-Mailer: gophish
Message-Id: <1619081069784230589.14601.5364162700216065009@demophish>
Subject: Tu cuenta de Outlook va a ser deshabilitada
To: Marta Barrio <demophish2019@gmail.com>
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: quoted-printable
```

4.
Part IV

How to make it **more professional**

Improvements
Headers

- The **Sender Policy Framework**, or **SPF**, is responsible for certifying which IPs can send mail using the domain in question. This record is effective against phishing attacks. It also helps destination servers to be more trustworthy and not to classify legitimate mail sent by you as SPAM.
- **DomainKeys Identified Mail**, or **DKIM**, is a registry that allows you to sign mail with your domain using public keys specified in your domain zones. In this way, the recipient is sure that the mail has been sent from your server and has not been intercepted and/or forwarded from another unauthorised server.
- **Domain-based Message Authentication, Reporting and Conformance**, or **DMARC**, which complements SPF and DKIM. This log indicates what to do when the previous logs fail, so that you can take the necessary measures as soon as possible.

https://dinahosting.com/ayuda/que-son-los-registros-spf-dkim-y-dmarc/

Improvements
Deployment

```
root@let:~/go/src/github.com/gophish/gophish# rg X-Gophish
webhook/webhook.go
29:     SignatureHeader = "X-Gophish-Signature"

models/email_request.go
122:            msg.SetHeader("X-Gophish-Contact", conf.ContactAddress)

models/maillog_test.go
234:            "X-Gophish-Contact": s.config.ContactAddress,
246:            "X-Gophish-Contact": "",
254:                    Header{Key: "X-Gophish-Contact", Value: ""},

models/maillog.go
186:            msg.SetHeader("X-Gophish-Contact", conf.ContactAddress)

models/email_request_test.go
82:             "X-Gophish-Contact": s.config.ContactAddress,
```

```
root@let2:~/go/src/github.com/gophish/gophish# rg X-Gophish
webhook/webhook.go.bak.bak
29:     SignatureHeader = "X-Gophish-Signature"

webhook/webhook.go.bak
29:     SignatureHeader = "X-Gophish-Signature"

models/email_request.go.bak.bak
122:            msg.SetHeader("X-Gophish-Contact", conf.ContactAddress)

models/maillog_test.go.bak.bak
234:            "X-Gophish-Contact": s.config.ContactAddress,
246:            "X-Gophish-Contact": "",
254:                    Header{Key: "X-Gophish-Contact", Value: ""},

models/maillog.go.bak.bak
186:            msg.SetHeader("X-Gophish-Contact", conf.ContactAddress)
```

```
find . -type f -exec sed -i.bak 's/X-Gophish-Contact/X-Contact/g' {} +
find . -type f -exec sed -i.bak 's/X-Gophish-Signature/X-Signature/g' {} +
```

Improvements
Deployment

```
root@let:~/go/src/github.com/gophish/gophish# rg -B1 ServerName config/config.go
43-
44:// ServerName is the server type that is returned in the transparency response.
45:const ServerName = "gophish"
```

```
root@let:~/go/src/github.com/gophish/gophish# rg -B1 ServerName config/config.go
43-
44:// ServerName is the server type that is returned in the transparency response.
45:const ServerName = "IGNORE"
```

Improvements

Deployment

```go
// Overwrite go's internal not found to allow templating the not found page
// The templating string is currently not passed in, therefore there is no templating yet
// If I need it in the future, it's a 5 minute change...
func customNotFound(w http.ResponseWriter, r *http.Request) {
	tmpl404, err := template.ParseFiles("templates/404.html")
	if err != nil {
		log.Fatal(err)
	}
	var b bytes.Buffer
	err = tmpl404.Execute(&b, "")
	if err != nil {
		http.NotFound(w, r)
		return
	}
	customError(w, b.String(), http.StatusNotFound)
}
```

```
root@let:~/go/src/github.com/gophish/gophish/templates# cat 404.html
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>KavaCON 2021</center>
</body>
</html>
```

Improvements
Deployment

```go
// Overwrite net.https Error with a custom one to set our own headers
// Go's internal Error func returns text/plain so browser's won't render the html
func customError(w http.ResponseWriter, error string, code int) {
        w.Header().Set("Content-Type", "text/html; charset=utf-8")
        w.Header().Set("X-Content-Type-Options", "nosniff")
        w.WriteHeader(code)
        fmt.Fprintln(w, error)
}
```

```go
// RecipientParameter is the URL parameter that points to the result ID for a recipient.
const RecipientParameter = "rid"
```

sed -i 's/const RecipientParameter = "rid"/const RecipientParameter = "cod"/g' models/campaign.go

```go
// RecipientParameter is the URL parameter that points to the result ID for a recipient.
const RecipientParameter = "cod"
```

Improvements
Deployment

```
root@let:~/go/src/github.com/kgretzky/evilginx2# grep -Ri 'X-Evilginx' *
root@let:~/go/src/github.com/kgretzky/evilginx2#
```

```
# check what we are about to remove
sed -n -e '183p;350p;377,379p;381p;407p;562,566p;580p;1456,1463p' core/http_proxy.go

# remove + backup original
sudo sed -i.bak -e '183d;350d;377,379d;381d;407d;562,566d;580d;1456,1463d' core/http_proxy.go
```

https://cilynx.com/how-to/evilginx2-vs-2fa-phishing/424/

```
  GNU nano 4.8                                                    core/config.go
const DEFAULT_REDIRECT_URL = "https://www.youtube.com/watch?v=dQw4w9WgXcQ" // Rick'roll
```

```
  GNU nano 4.8                                                    core/config.go
const DEFAULT_REDIRECT_URL = "https://miralaurl.es/login" // OK
```

5.
References

Documentation and resources

References

- https://github.com/kgretzky
- https://breakdev.org/evilginx-2-next-generation-of-phishing-2fa-tokens/
- https://breakdev.org/evilginx-2-3-phishermans-dream/
- https://breakdev.org/evilginx-2-4-gone-phishing/
- https://www.youtube.com/watch?v=QRyinxNY0fk
- https://getgophish.com/documentation/
- https://medium.com/@valeriyshevchenko/how-to-perform-phishing-attack-with-2fa-e9d633c66383
- https://www.sprocketsecurity.com/blog/never-had-a-bad-day-phishing-how-to-set-up-gophish-to-evade-security-controls
- https://www.blackhillsinfosec.com/webcast-how-to-build-a-phishing-engagement-coding-ttps/

Thanks!

Any questions?

You can find us at @martrudix & @Carol12Gory