Jeopardize

a low(zero) cost threat intelligence&response tool against phishing domains

Utku Sen - DEF CON 28 Demo Labs
Who am I

- Security Analyst @HackerOne
- utkusen.com
- github.com/utkusen
- twitter.com/utkusen
Hello! This is a totally legitimate email and you have nothing to worry about! Now, please click the link below and type in your password!
Paypal Christmas Gifts
@PaypalChristmasGifts

paypall-christmasgifts.com

log onto your account, verify your details, for your chance to be in Paypal’s new year draw.

PayPal

5:10 PM · Jan 1, 2019 · Twitter Web Client

Promoted by Paypal Christmas Gifts

1 Retweet 7 Likes
Attack Timeline

• 10:00 - Attacker promotes phishing tweet
• 10:15 - 300 people has seen the tweet
• 10:20 - 5 people entered their credentials
• 10:25 - accounts compromised
• 10:30 - Tweet is taken down due to the abuse reports
Solution?

- Abuse complaint to the domain&host provider ❌
- Abuse complaint to the Twitter ❌
- Abuse complaint to the Google ❌
- Be more proactive ✔
Being Proactive

- Detect possible phishing domains before the campaign ✔️
- Do DDOS ❌ (If you have a huge budget) ✔️
- Confuse the attacker ✔️
- Observe the honey token flow ✔️
What “jeopardize” does?
Early Detection

- Detects typosquatting, homograph etc. domains (fcaebbook.com)
- **Analyzes**: web server, name servers, HTML forms, Alexa ranking, page size, registration date etc.
- Provides a “Risk Score”

<table>
<thead>
<tr>
<th>Domain</th>
<th>NS</th>
<th>Web Server</th>
<th>Login Form</th>
<th>Registration</th>
<th>Risk Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCAEBOOK.com</td>
<td>Cloudflare</td>
<td>✅</td>
<td>✅</td>
<td>New</td>
<td>90</td>
</tr>
<tr>
<td>FACEBOOKADS.com</td>
<td>Godaddy</td>
<td>✅</td>
<td>❌</td>
<td>Old</td>
<td>50</td>
</tr>
</tbody>
</table>
Confusing the Attacker & Buying Time

Attacker

http://twitter.com

AcmeBank TaxRefund

http://acmebnak.com

Username: password:

Sends Fake & Honey Creds

User

Clicks

Jeopardize

<table>
<thead>
<tr>
<th>Username</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utku</td>
<td>Kralbesiktas</td>
</tr>
<tr>
<td>David</td>
<td>Kingwestham</td>
</tr>
<tr>
<td>Serdar</td>
<td>Tisssss</td>
</tr>
<tr>
<td>John</td>
<td>Lol123</td>
</tr>
</tbody>
</table>

LEGIT

FAKE&HONEY

FAKE&HONEY

FAKE&HONEY
From The Attacker’s Perspective

- Possibility 1) Most of the credentials are not working
- Possibility 2) All credentials are working but most of them are empty
- Losing time
From The Defender’s Perspective

- Observing the usage ratio of honey tokens.
- Tracking the attacker’s IP easily
- Detecting affected users
- More time to take precautions
How “jeopardize” works
Generating Domain Combinations

acmebank
  └── amcebank

acmebank
  ├── acmebnak
  │    └── Acmeban
  └── acmabank
Detecting Registered Domains

• **Brute:** Combine the generated words (acmebnak, amcebank etc.) with all TLDs (com, net, xyz, live etc.)
  - **Pros:** Free
  - **Cons:** Slow

• **Daily:** Search the generated words (acmebnak, amcebank etc.) in daily registered domains
  - **Pros:** Fast
  - **Cons:** Requires zonefiles.io API key
Analyzing the Domains

- IP
- Web Server
- Name Servers
- Page Size
- If login form exists
- SSL Certificate
- Registration Date
- Alexa Ranking
Jeopardizing the Login Forms

- Feed the jeopardize with honey tokens
- Jeopardize sends them to the target domains
Saving the Results

```
<domain>
  <address>acmebnak.com</address>
  <name_servers>ns1.cloudflare.com ns2.cloudflare.com</name_servers>
  <mx_servers> </mx_servers>
  <date_flag>True</date_flag>
  <alexa_flag>False</alexa_flag>
  <webserver_flag>True</webserver_flag>
  <certificate_flag>False</certificate_flag>
  <form_flag>True</form_flag>
  <phishing_score>85</phishing_score>
</domain>
```
Usage
python3 jeopardize.py --domain facebook.com --type brute
python3 jeopardize.py --domain facebook.com --type incremental
python3 jeopardize.py --domain facebook.com --type daily -U user.txt -P pass.txt
https://github.com/utkusen/jeopardize