WISE PHISHERMEN NEVER TRUST THE WEATHER

TACKLING MODERN REVERSE PROXY ANTI-PHISHING PROTECTIONS

Kuba Gretzky

@mrgretzky

00 // WHOAMI

KUBA GRETZKY

Offensive Security Tools Developer

Ex Game Hacker

breakdev.org - offensive security blog

EVILGINX - MFA phishing framework (2017)

BREAKDEV RED - community for red teamers

Evilginx Mastery - Evilginx phishing course

EVILGINX PRO (2025)

@mrgretzky

01 // INTRO

WHAT AM I DOING?

- **EVILGINX PRO** - launched on February, 25th 2025 *(finally!)*
- Offload excess tasks from the red teams
- Preserving the regional phishing tradition
- **BREAKDEV RED** - registration & manual approval
- Decided to develop a shop engine from scratch *(took ~6 months)*

BREAKDEV RED

APPLICATION FORM

Please fill out the application form below to begin the verification process for joining the community:

John

Doe

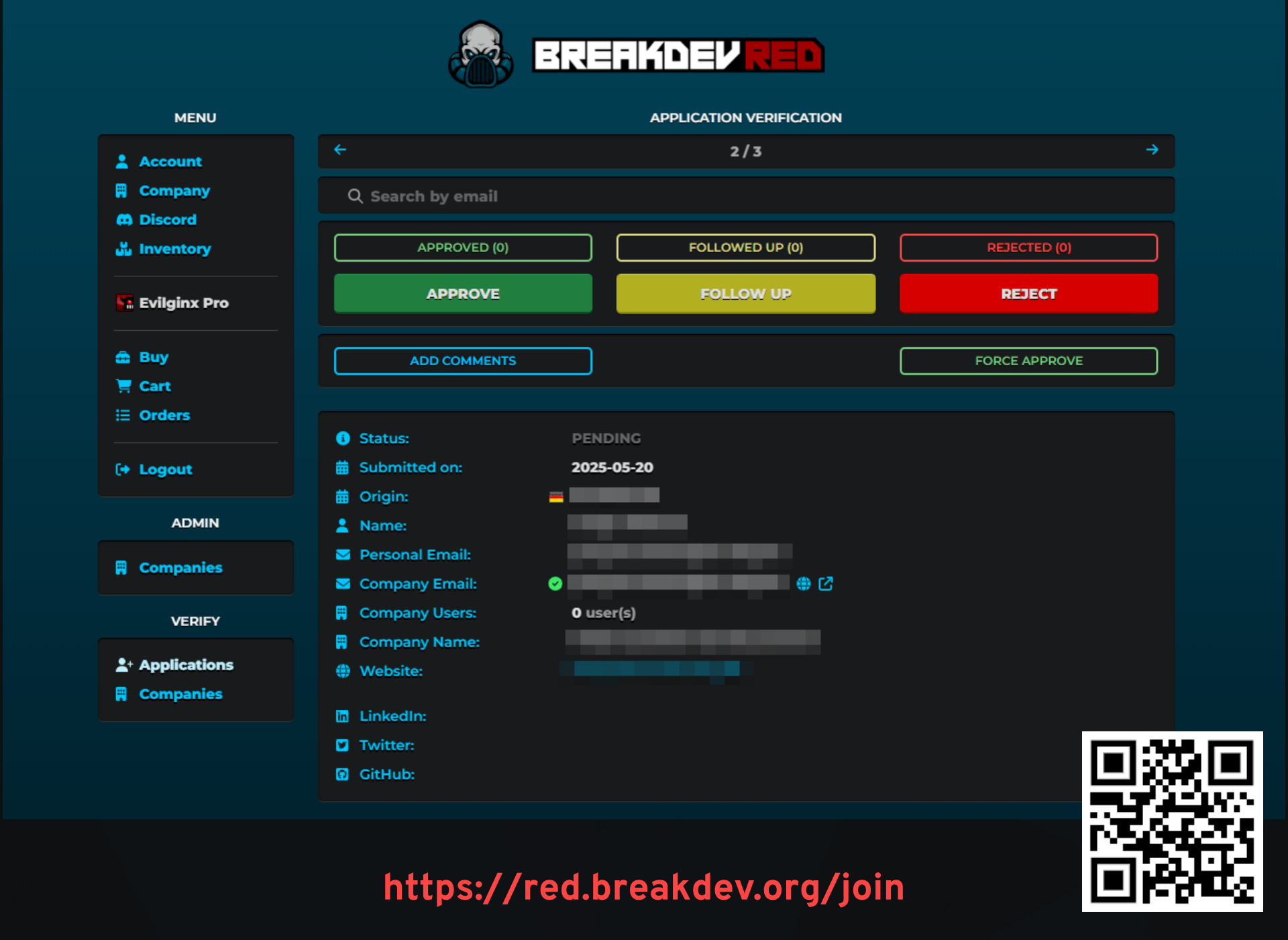
john.doe@gmail.com

jd@haxorteam.com

Haxor Team

https://www.haxorteam.org

https://red.breakdev.org/join


BREAKDEV RED
MENU
Account
Company
Discord
Inventory
Evilginx Pro
Buy
Cart
Orders
Logout
ADMIN
Companies
VERIFY
Applications
Companies
APPLICATION VERIFICATION
2 / 3
Search by email
APPROVED (0)
FOLLOWED UP (0)
REJECTED (0)
APPROVE
FOLLOW UP
REJECT
ADD COMMENTS
FORCE APPROVE
Status:
PENDING
Submitted on:
2025-05-20
Origin:
Name:
Personal Email:
Company Email:
Company Users:
0 user(s)
Company Name:
Website:
LinkedIn:
Twitter:
GitHub:
https://red.breakdev.org/join

MENU

- Account
- Company
- Discord
- Inventory
- Evilginx Pro
- Buy
- Cart
- Orders
- Logout

ADMIN

- Companies

VERIFY

- Applications
- Companies

INVENTORY

Name	Assigned to	Valid from	Expires on	
Evilginx Pro 12-month subscription	kuba@breakdev.org	2024-12-19	2025-12-19 213 days left	RENEW

CREATE A NEW ACCOUNT

https://red.breakdev.org/join

01 // INTRO

WHAT AM I DOING?

- **EVILGINX PRO** - launched on February, 25th 2025 *(finally!)*
- Motivation: Offload excess tasks from the red teams
- **BREAKDEV RED** - registration & manual approval
- Decided to develop a shop engine from scratch *(took ~6 months)*
- Approved 1700+ hackers into the community

Shop engine made from scratch

+

1700+ hackers

=

01 // INTRO

WHAT IS THE TALK ABOUT?

- State of **reverse proxy** phishing detections in 2025
- I'll be focusing on **phishing page detection** rather than emails
- **How to evade** modern anti-phishing protections?
  (like Google Chrome Safe Browsing)

02 // THREE LAYERS OF DECEPTION

CORE ANTI-PHISHING EVASION TACTICS

STEALTH:

- TLS certificate gets registered *(TLS Transparency Report)*
- Prevent public exposure of the phishing hostname

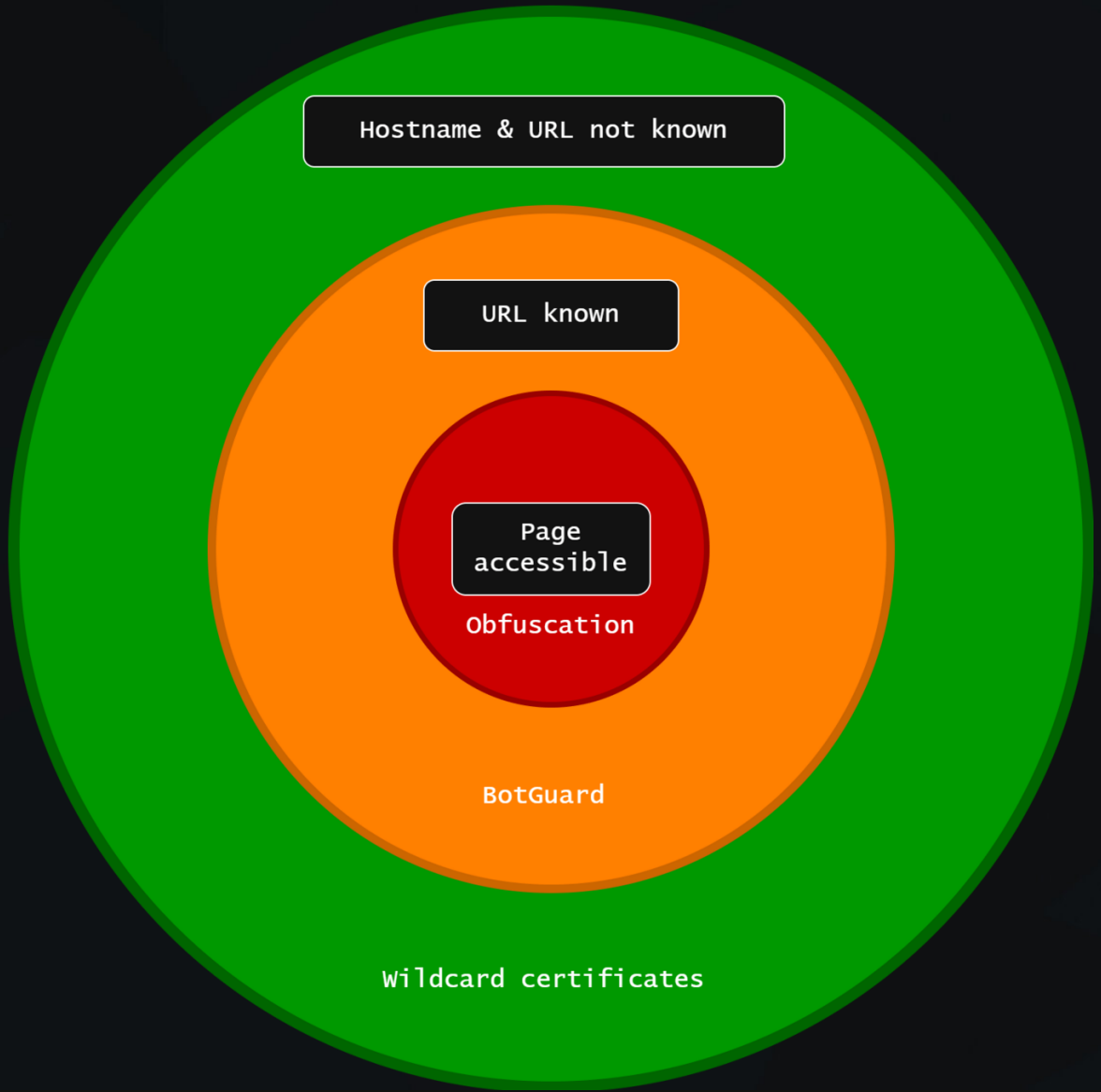
DECEPTION:

- The phishing URL gets exposed
- Prevent automated scanners from reaching the phishing page

OBFUSCATION:

- The phishing page gets exposed
- Prevent automated scanners from detecting a phishing attempt

02 // THREE LAYERS OF DECEPTION



evilginxpro

03 // STEALTH

PROTECTING THE PHISHING HOSTNAME

- Every phishing page needs a TLS certificate to support HTTPS
- Every registered TLS certificate is made public
- TLS Transparency Report browser: https://crt.sh
- Hostname resolves to HTTPS server's IP address
- HTTPS server can be analyzed

03 // STEALTH

NON-WILDCARD TLS CERTIFICATES

crt.sh Identity Search

Group by Issuer

Criteria	Type: Identity	Match: ILIKE	Search: 'evilginx.com'

crt.sh ID	Logged At ⇧	Not Before	Not After	Common Name	Matching Identities	Issuer Name
18125175993	2025-04-29	2025-04-29	2025-07-28	qr.evilginx.com	qr.evilginx.com	C=US, O=Let's Encrypt, CN=E5
18125721190	2025-04-29	2025-04-29	2025-07-28	qr.evilginx.com	qr.evilginx.com	C=US, O=Let's Encrypt, CN=E5
18125175404	2025-04-29	2025-04-29	2025-07-28	lab.evilginx.com	lab.evilginx.com	C=US, O=Let's Encrypt, CN=E6
18125717833	2025-04-29	2025-04-29	2025-07-28	lab.evilginx.com	lab.evilginx.com	C=US, O=Let's Encrypt, CN=E6
18125614449	2025-04-29	2025-04-29	2025-07-28	deusex.lab.evilginx.com	deusex.lab.evilginx.com	C=US, O=Let's Encrypt, CN=E5
18125715885	2025-04-29	2025-04-29	2025-07-28	deusex.lab.evilginx.com	deusex.lab.evilginx.com	C=US, O=Let's Encrypt, CN=E5
18123459863	2025-04-29	2025-04-29	2025-07-28	cyberpunk.lab.evilginx.com	cyberpunk.lab.evilginx.com	C=US, O=Let's Encrypt, CN=E5
18127020308	2025-04-29	2025-04-29	2025-07-28	cyberpunk.lab.evilginx.com	cyberpunk.lab.evilginx.com	C=US, O=Let's Encrypt, CN=E5
18125166647	2025-04-29	2025-04-29	2025-07-28	bladerunner.lab.evilginx.com	bladerunner.lab.evilginx.com	C=US, O=Let's Encrypt, CN=E6
18125757808	2025-04-29	2025-04-29	2025-07-28	bladerunner.lab.evilginx.com	bladerunner.lab.evilginx.com	C=US, O=Let's Encrypt, CN=E6
18126993383	2025-04-29	2025-04-29	2025-07-28	akira.lab.evilginx.com	akira.lab.evilginx.com	C=US, O=Let's Encrypt, CN=E6
18126991346	2025-04-29	2025-04-29	2025-07-28	akira.lab.evilginx.com	akira.lab.evilginx.com	C=US, O=Let's Encrypt, CN=E6
17701340363	2025-04-06	2025-04-06	2025-07-05	evilginx.com	evilginx.com	C=US, O=Let's Encrypt,

03 // STEALTH

NON-WILDCARD TLS CERTIFICATES

367308017	2018-03-27	2018-03-27	2018-06-25	goog-demo.evilginx.com	accounts.goog-demo.evilginx.com goog-demo.evilginx.com ssl.goog-demo.evilginx.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Auth X3
299435251	2018-01-11	2018-01-11	2018-04-11	live.evilginx.com	live.evilginx.com login.live.evilginx.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Auth X3
299435301	2018-01-11	2018-01-11	2018-04-11	dropbox.evilginx.com	dropbox.evilginx.com www.dropbox.evilginx.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Auth X3
299434925	2018-01-11	2018-01-11	2018-04-11	google.evilginx.com	accounts.google.evilginx.com google.evilginx.com ssl.google.evilginx.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Auth X3
299434819	2018-01-11	2018-01-11	2018-04-11	icloud.evilginx.com	appcdn.icloud.evilginx.com appleid.icloud.evilginx.com edge.icloud.evilginx.com icloud.evilginx.com icloud.icloud.evilginx.com id.icloud.evilginx.com idmsa.icloud.evilginx.com setup.icloud.evilginx.com www.icloud.evilginx.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Auth X3
299435071	2018-01-11	2018-01-11	2018-04-11	linkedin.evilginx.com	linkedin.evilginx.com www.linkedin.evilginx.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Auth X3
299434196	2018-01-11	2018-01-11	2018-04-11	facebook.evilginx.com	facebook.evilginx.com m.facebook.evilginx.com www.facebook.evilginx.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Auth X3
145427967	2017-05-28	2017-05-28	2017-08-26	icloud.evilginx.com	appcdn.icloud.evilginx.com appleid.icloud.evilginx.com edge.icloud.evilginx.com icloud.evilginx.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Auth X3

03 // STEALTH


ONE DOES NOT SIMPLY
REMOVE TLS CERTIFICATES FROM THE INTERNET
imgflip.com

03 // STEALTH

WILDCARD TLS CERTIFICATES

18181444539	2025-05-02	2025-05-02	2025-07-31	*.signin.linkedin.phish4.life	*.signin.linkedin.phish4.life
18181450357	2025-05-02	2025-05-02	2025-07-31	*.signin.linkedin.phish4.life	*.signin.linkedin.phish4.life
18180474580	2025-05-02	2025-05-02	2025-07-31	*.signin.phish4.life	*.signin.phish4.life
18180482750	2025-05-02	2025-05-02	2025-07-31	*.signin.phish4.life	*.signin.phish4.life
18145576430	2025-04-30	2025-04-30	2025-07-29	*.its.my.live.phish4.life	*.its.my.live.phish4.life
18145564557	2025-04-30	2025-04-30	2025-07-29	*.its.my.live.phish4.life	*.its.my.live.phish4.life
18126739702	2025-04-29	2025-04-29	2025-07-28	*.m365.phish4.life	*.m365.phish4.life
18129319107	2025-04-29	2025-04-29	2025-07-28	*.m365.phish4.life	*.m365.phish4.life
18040497224	2025-04-24	2025-04-24	2025-07-23	*.ms.phish4.life	*.ms.phish4.life
18040496458	2025-04-24	2025-04-24	2025-07-23	*.ms.phish4.life	*.ms.phish4.life
18040218319	2025-04-24	2025-04-24	2025-07-23	*.boop.phish4.life	*.boop.phish4.life
18040226970	2025-04-24	2025-04-24	2025-07-23	*.boop.phish4.life	*.boop.phish4.life
17979347643	2025-04-21	2025-04-21	2025-07-20	*.okta.phish4.life	*.okta.phish4.life
17979356404	2025-04-21	2025-04-21	2025-07-20	*.okta.phish4.life	*.okta.phish4.life

03 // STEALTH

WILDCARD TLS CERTIFICATES

TAKEAWAYS:

- Scanners need to guess the subdomain name *(dictionary attack?)*
- Unable to resolve the HTTPS server's IP address
- Unable to connect to the HTTPS server, not knowing the hostname

04 // DECEPTION

PREVENTING ACCESS TO THE PHISHING PAGE

- The phishing URL eventually gets exposed when sent to the target
- Security products will open the phishing URL & scan the page
- How can we prevent bots from seeing the phishing page?

04 // DECEPTION

Web browser

Proxied
Spoofed Website
(hidden page body)

Proxied
Login Page

Evilginx Pro

Spoofed website
(https://www.wikipedia.org)

Target website
(https://accounts.google.com)

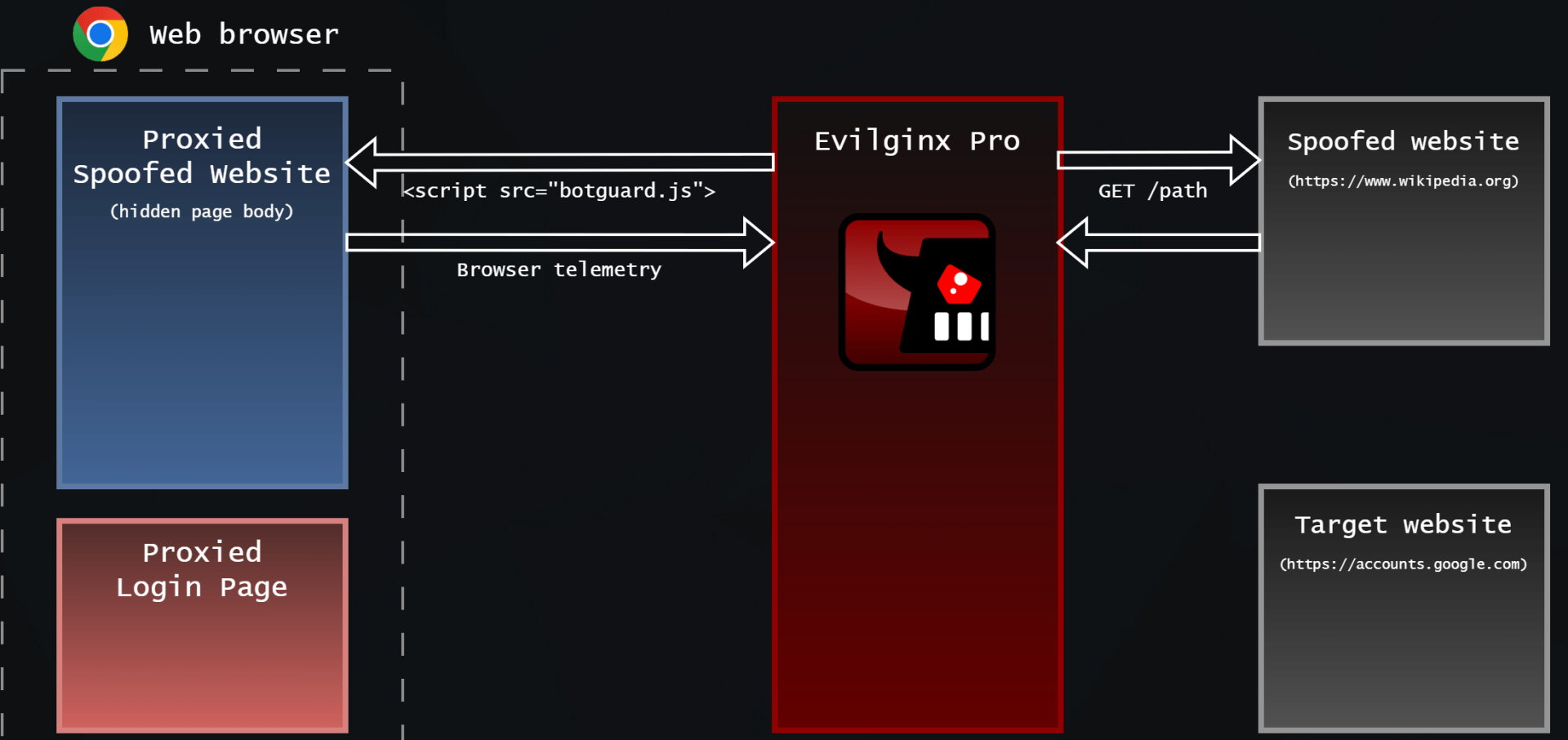
evilginxpro

04 // DECEPTION

04 // DECEPTION

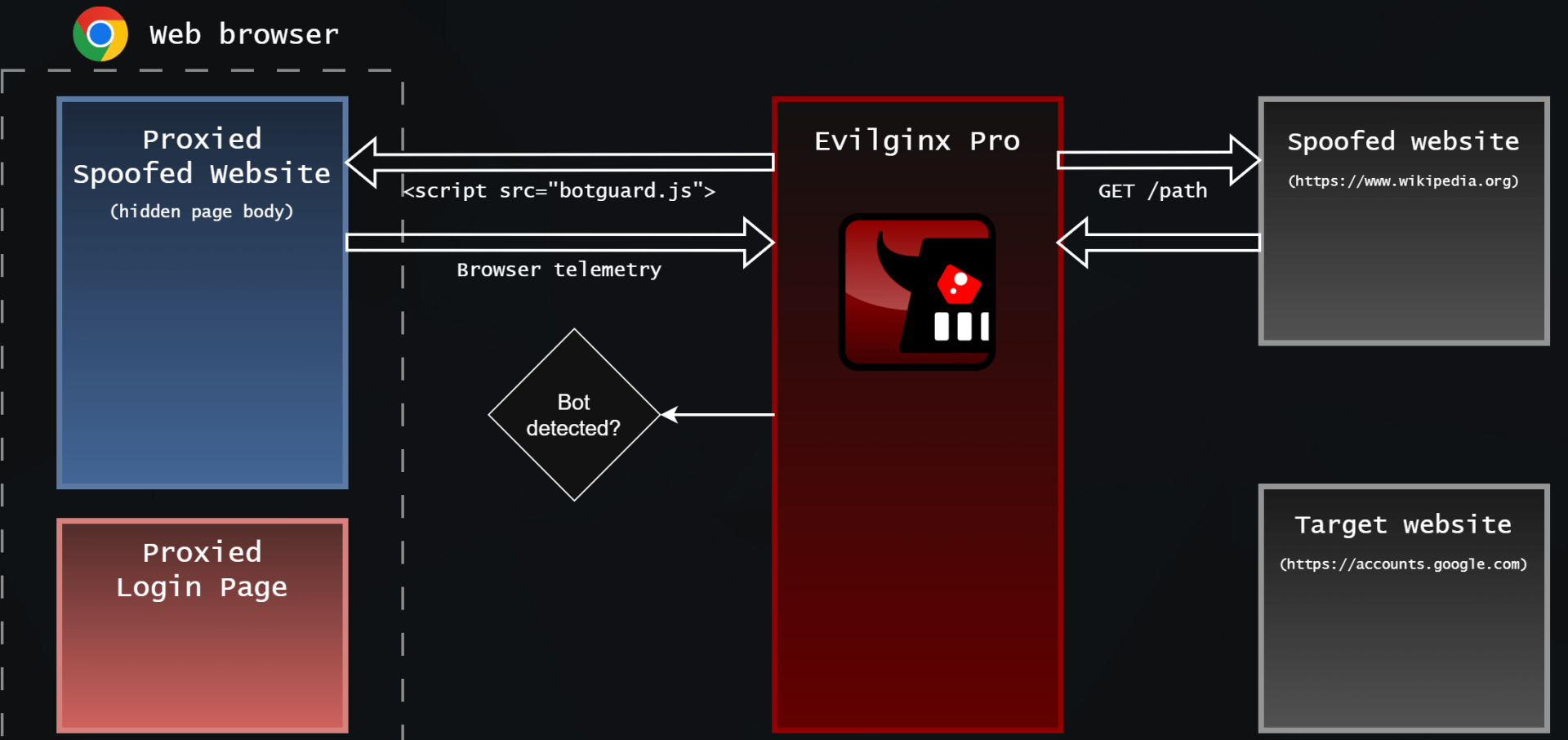
evilginxpro

04 // DECEPTION



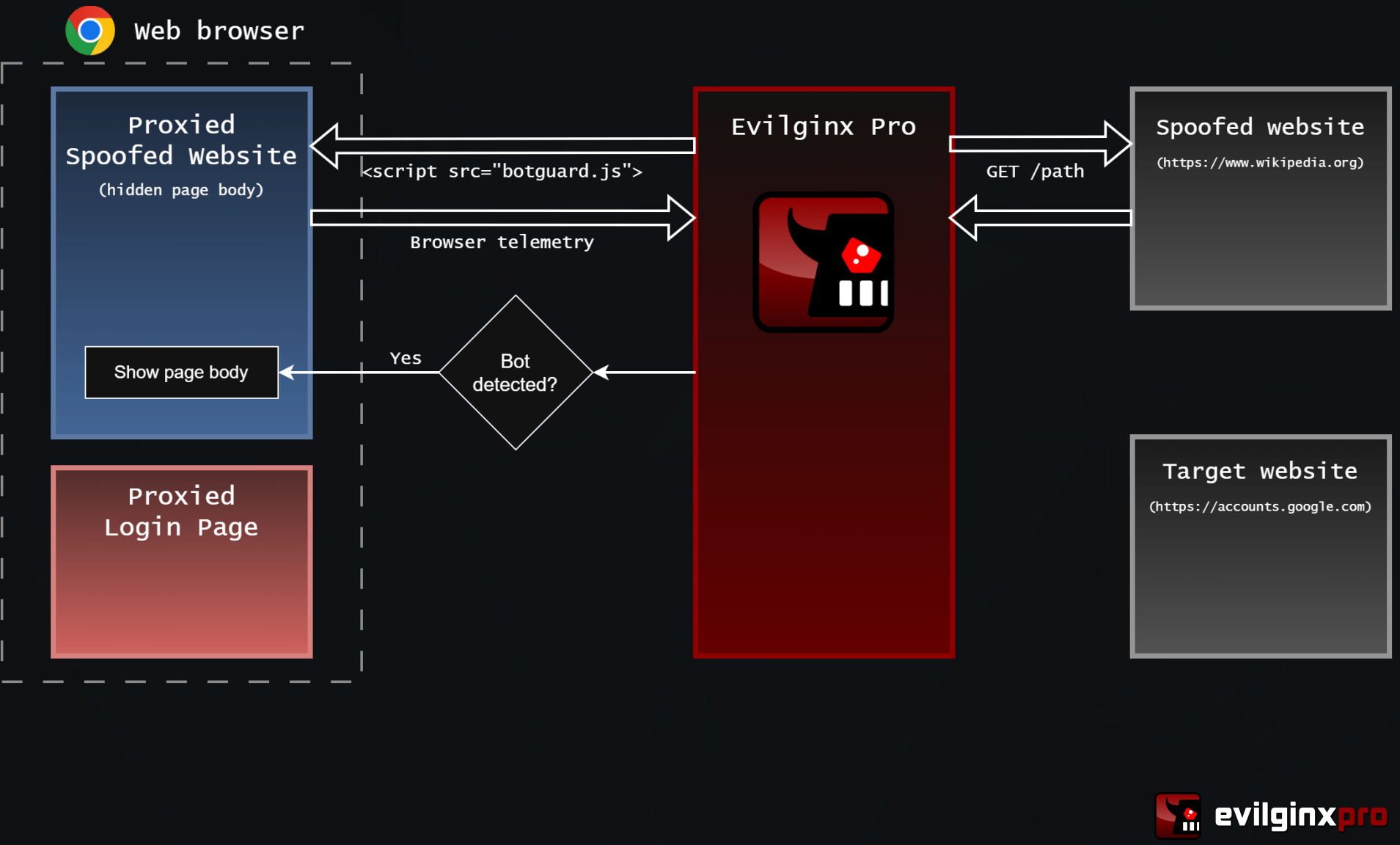
evilginxpro

04 // DECEPTION



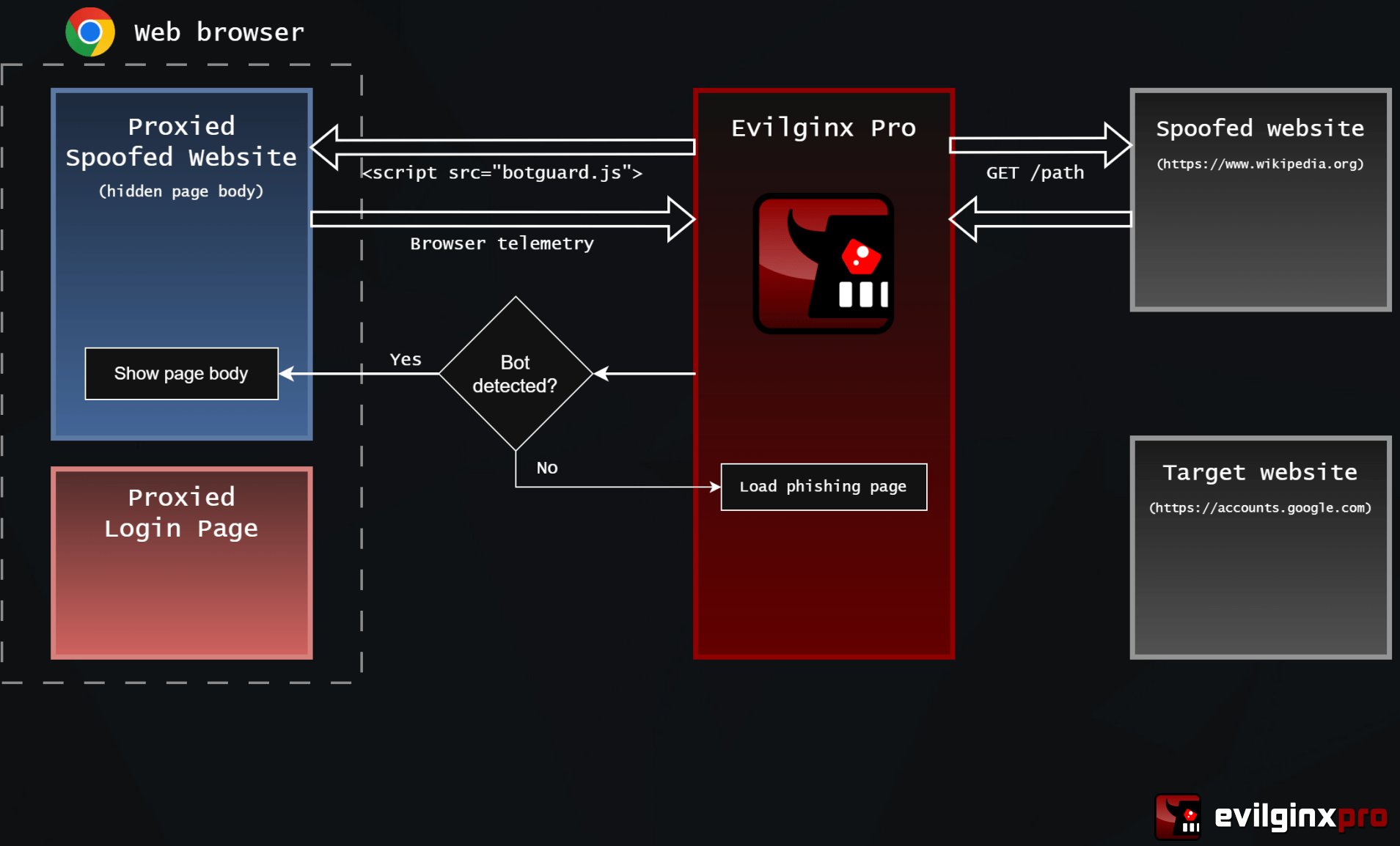
evilginxpro

04 // DECEPTION



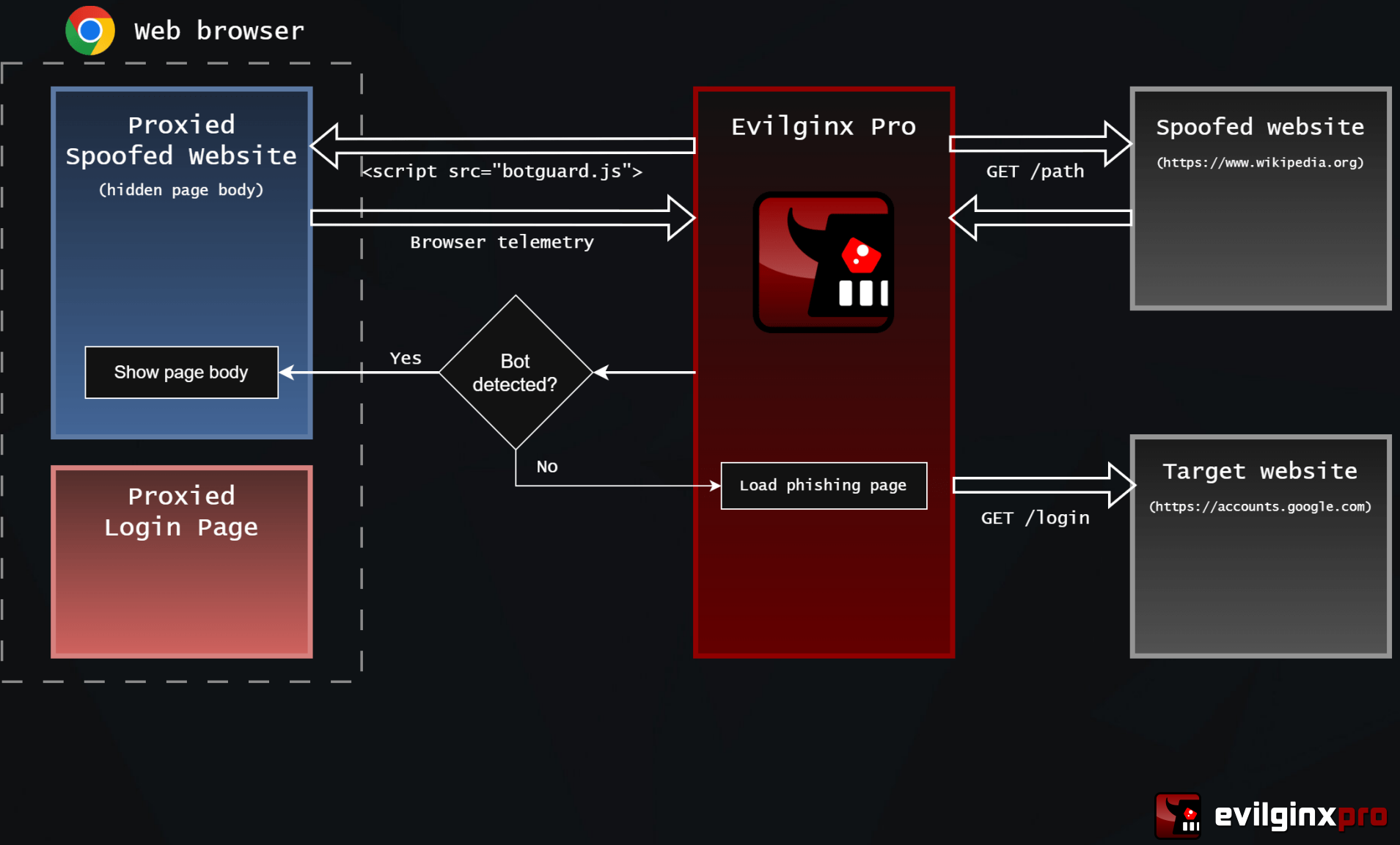
evilginxpro

04 // DECEPTION



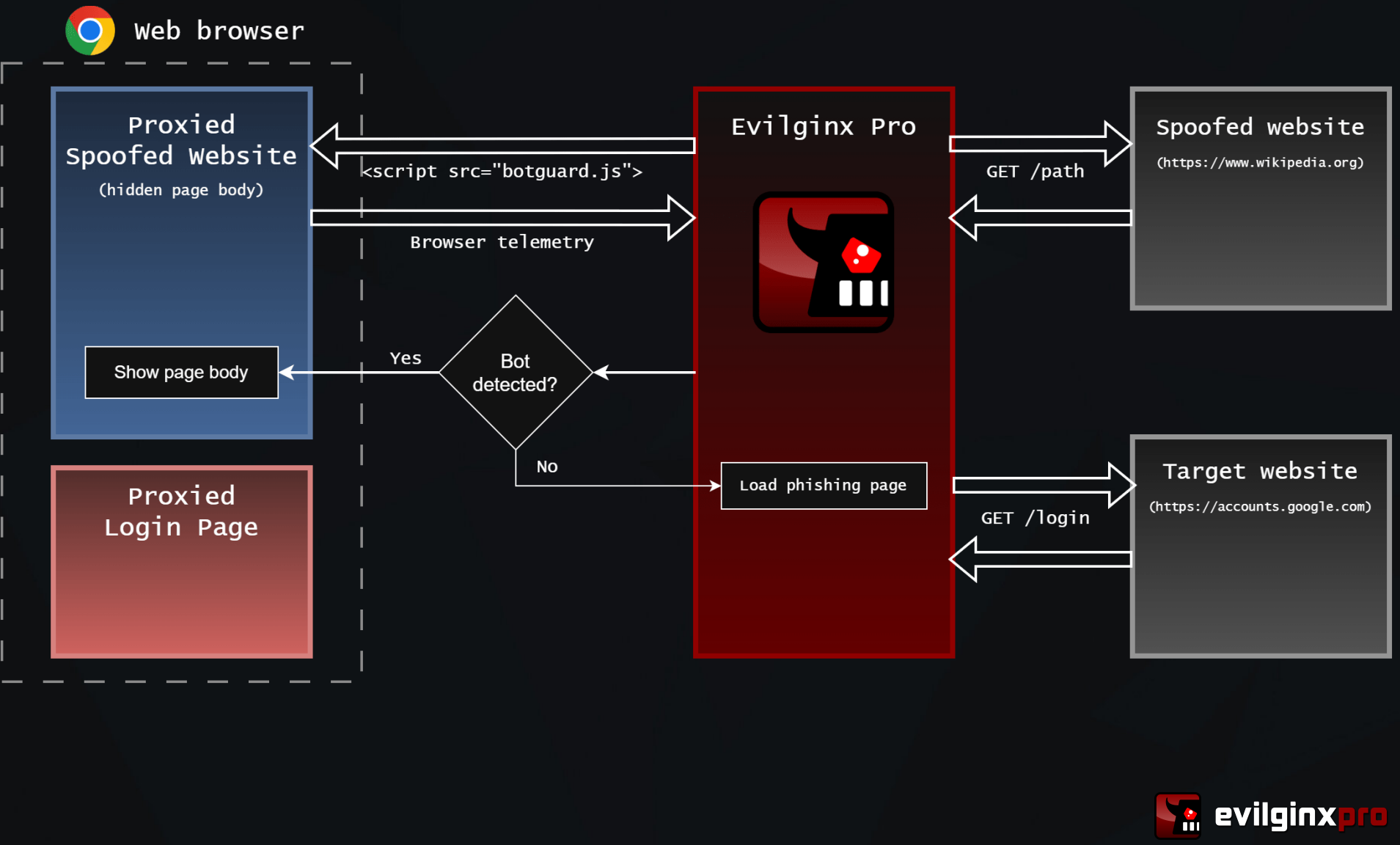
evilginxpro

04 // DECEPTION

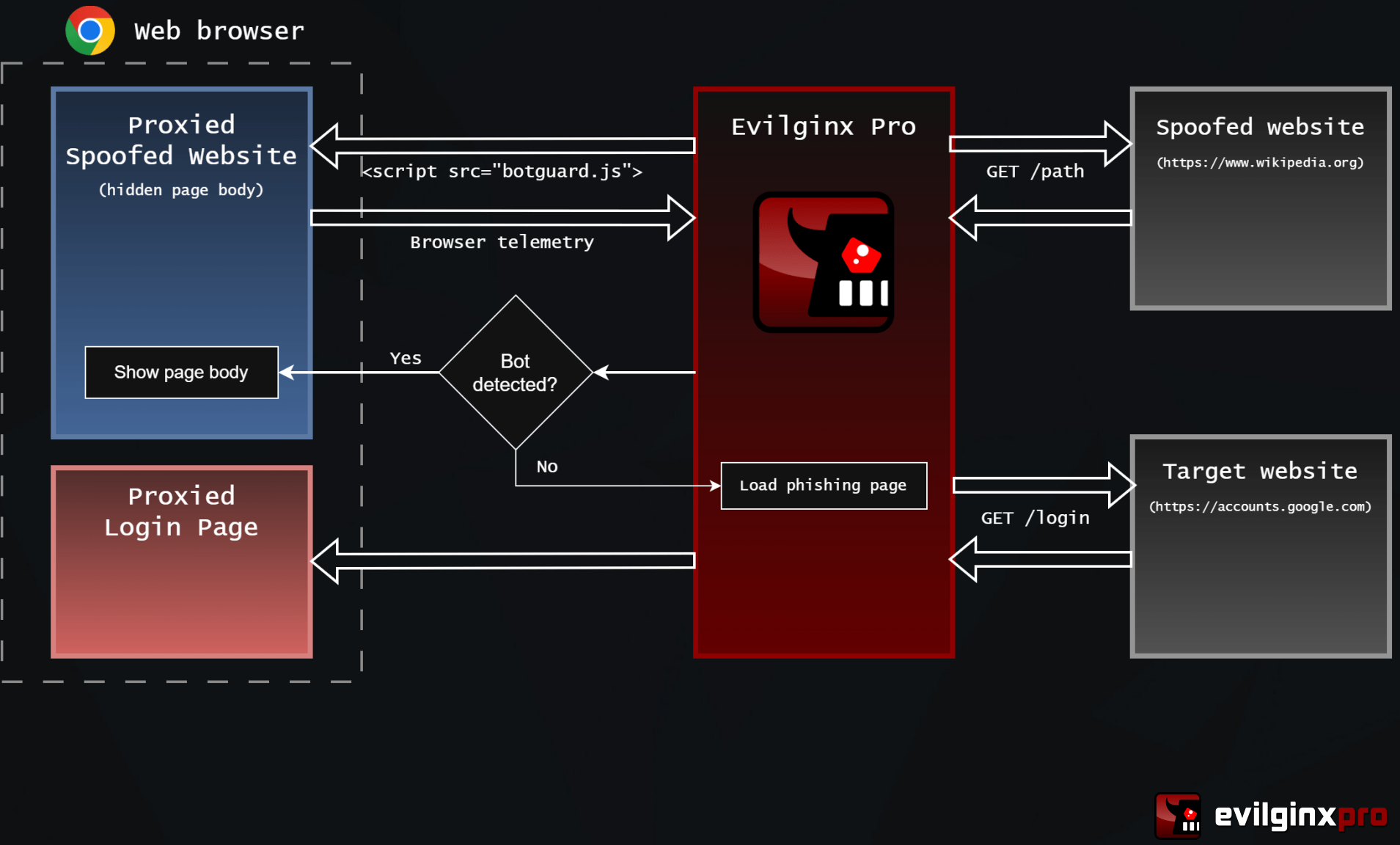


evilginxpro

04 // DECEPTION



04 // DECEPTION



evilginxpro

04 // DECEPTION

PREVENTING ACCESS TO THE PHISHING PAGE

TAKEAWAYS:

- The majority of bots do not have JavaScript enabled
- Bots see the proxied content of a legitimate website *(spoofing)*
- Human visitors get redirected to the phishing page
- BotGuard - supposed to be a cheap knockoff of Cloudflare Turnstile
- BotGuard - allegedly bypasses Microsoft Defender for Endpoint (MDE)

05 // OBFUSCATION

HAMPERING THE ANALYSIS OF THE PHISHING PAGE

DETECTION TOOLS:

- Google Chrome Safe Browsing
- Browser extensions *(e.g. Push Security)*
- Canary Tokens *(e.g. Thinkst Canary)*

SCANNING TARGETS:

- HTML content
- JavaScript content
- Request URLs
- DOM structure

05 // OBFUSCATION

HTML OBFUSCATION

```
<!-- Copyright (C) Microsoft Corporation. All rights reserved. -->
<!DOCTYPE html>
<html dir="ltr" class="" lang="en">
<head>
    <title>Sign in to your account</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=2.0, user-scalable=yes">
    <meta http-equiv="Pragma" content="no-cache">
    <meta http-equiv="Expires" content="-1">
    <link rel="preconnect" href="https://aadcdn.msauth.net" crossorigin>
<meta http-equiv="x-dns-prefetch-control" content="on">
<link rel="dns-prefetch" href="//aadcdn.msauth.net">
<link rel="dns-prefetch" href="//aadcdn.msftauth.net">

    <meta name="PageID" content="ConvergedSignIn" />
    <meta name="SiteID" content="" />
    <meta name="ReqLC" content="1033" />
    <meta name="LocLC" content="en-US" />


        <meta name="format-detection" content="telephone=no" />

    <noscript>
        <meta http-equiv="Refresh" content="0; URL=https://login.microsoftonline.com/jsdisabled" />
    </noscript>



<meta name="robots" content="none" />

<script type="text/javascript" nonce='5GLwl5UTFUBm8pMmDFip1A'>//<![CDATA[
$Config={"fShowPersistentCookiesWarning":false,"urlMsaSignUp":"https://login.live.com/oauth20_authorize.srf?client_id=4765445b-32c6-49b0-
```

05 // OBFUSCATION

HTML OBFUSCATION

```
<!DOCTYPE html>
<script>document.write(atob('PCFET0NUWVBFIGh0bWw+PGh0bWwgbGFuZz0iZW4tVVMiIGR&#pcj0ibHRyIj48aGVhZ&#D48YmFzZSBocmV&#mPSJodHRwczovL2FjY291bn
RzLmJvb3Auc&#Ghpc2g0bGlmZS5jby51ay92My9zaWduaW4vIi8+PGxpbmsgcmVsPSJwcmVj&#b25uZWN0IiBocmVmPSIvL2dzdGF0aW&#MuYm9vcC5waGlzaDRsaWZlLmNvLnVrI
i8+PG&#1ldGEgbmFtZT0icmVmZXJyZXIiIGNvbnRlbnQ9Im9yaWdpbiI&#vPjxzY3JpcHQ&#gZGF0YS1pZD0iX2dkIiBub25jZT0iOEdJb0VkOWpm&#MW5Gc3NrY21jUXpKZyI+Zn
VuY3Rpb24gXzB4M2FiNyhfMHg0OTc5MGIsXzB4N&#jg3Mjgpe3ZhciBfMHg0MWEyOTQ9XzB4NDFhMigpO3JldHVybiBfMHgzYWI3PWZ1&#bmN0aW9uKF8weDNhYjdhZCxfMHg0NDh
mM2Qpe18weDNhYjdhZD&#1fMHgzYWI3YWQtMHgxNjI7dmFyIF8weDRjOTY&#yZD1fMHg0MWEyOTRbXzB4M2FiN2FkXTtpZihfMHgzYWI3WydXclhQUUInX&#T09PXVuZGVmaW5lZC
l7dmFyIF8weDQ3ODA3&#ZT1mdW5jdGlvbihfMHgzY2FhOGYpe3Zhci&#BfMHg0ZmI5NTA&#9J2FiY2RlZmdoaWprbG1ub&#3BxcnN0dX&#Z3eHl6QUJDREVGR0hJSktMTU5PUFFSU
1RVVldYWVowMTIzNDU2Nzg5Ky89Jzt2Y&#XIgXzB4NTliZTA2PScnLF8weDRmMmVlZj0nJztmb3IodmFyIF8we&#DU5ODA4ZT0weDAsXzB4MzBiMTU4LF8weDJjMzE2NixfMHgzMT
Y0OTY9MHgwO1&#8weDJjMzE2Nj1f&#MHgzY2FhO&#GZbJ2NoYXJ&#BdCddKF8weDMxNjQ5NisrKTt+XzB4MmM&#zMTY2JiYoXzB4MzBiMTU4PV8weDU5ODA4ZSUweDQ/XzB&#4MzB
iMTU4KjB4NDArXzB4MmMzMTY2Ol&#8weDJjMzE2NixfMHg1OTgwOGUrKyUweDQpP18weDU5YmUwNis9U3RyaW5nWy&#dmcm9tQ2hhckNvZGUnXSgweGZmJl8weDMwYjE1OD4+KC0w
e&#DIqXzB4NTk4MDhlJjB4NikpOjB4MCl7XzB4MmMzMTY2&#PV8weDRmYjk1MFsnaW5kZXhPZiddKF8weDJjMzE2Nik7fWZv&#cih2YXIg&#XzB4MzhjM2ExPTB4MCxfMHhhYjJiM
D1fMHg1OWJlMDZbJ2x&#lbmd0aCddO18weDM4Y&#zNhMTxfMHhhYjJiMD&#tfMHgzOGMzYTErKyl7XzB4NGYyZWVmK&#z0nJScrKCcwMCcrXzB4NTliZTA2WydjaGFyQ29kZUF0J1
0oXz&#B4MzhjM2ExKVsndG9TdHJpb&#mcnXSgweDEwKSlbJ3NsaWNlJ10oLTB4Mik7f&#XJldHVybiBkZWNvZGVVUklDb2&#1wb25lbnQoXzB4N&#GYyZWVmKTt9O3ZhciBfM&#Hg
0ZWIwMDg9ZnVuY3Rpb24oXz&#B4NDMxN&#TY1LF8weDMxMzJhMyl7dmFyIF&#8weDIxNjU1Mz&#1bXSxfMHgxMGI1MjI9MHg&#wLF8weDQ0ZGIyYixfMHgyY2U4ZmY9Jyc7XzB4ND
MxNTY1PV8weDQ3ODA3ZSh&#fMHg0MzE1NjUpO3ZhciBfMHgxNjFmNzE7Zm&#9yKF8weDE2MWY3M&#T0weDA7XzB4MTYxZ&#jcxPDB4MTAwO18weDE2MWY3MSsrKXt&#fMHgyMTY1N
TNbXz&#B4MTYxZjcxXT1fMHgxNjFmNzE7&#fWZvcihfMHgxNjF&#mNzE9MHgwO18w&#eDE2MWY3MTwweDEwMDtfMHgxNjFmNzErKyl7XzB4MTBiNT&#IyPShfMHgxMGI1MjIrXzB4
MjE2NTUzW18weDE2MWY3MV0rXzB4MzEzM&#mEzWydjaGFyQ29kZUF0J10&#oXzB4MTYxZjcxJV8w&#eDMxMzJhM1&#snbGVuZ3RoJ10pKS&#UweDEwMCxfMHg0NGRiMmI9XzB4MjE
2NTUzW18weDE2MWY3MV0sXzB&#4MjE2NTUzW18weDE2MWY3MV09XzB4Mj&#E2NTUzW18weDEwYjUyMl0sXzB4MjE2NTUzW18weDEwYjUy&#Ml09XzB4NDRkYjJiO31fMHgxNjFmNz
E9MHgwLF8weDEwYjUyMj0weD&#A7Zm9yKHZhciBfMHg1Y2QzZmM9&#MHgwO18weDVjZDNmYzxfMHg0MzE1NjVbJ2x&#lbmd0aCddO&#18weDVjZDNmYysr&#KXtfMHgxNjFmNzE9K
F8weDE2MWY3MSsweDEpJ&#TB4MTAwLF8weDEwYjUyMj0oXzB4MTBiNTIyK18weDIxNjU1M1tfMHgxNj&#FmNzFdKSUweDEwMCxfMHg0NGRiMmI9XzB4MjE2NTUzW18weDE2MWY3MV
&#0sXzB4MjE2&#NTUzW18weDE2MWY3MV09XzB4MjE2NTU&#zW18weDEwYjUyMl0sXzB4M&#jE2NTUzW18weDEwYjUyMl09XzB4NDRkYj&#JiLF8weDJjZThmZis9U3RyaW5nWydmc
&#m9tQ2hhckNvZGUnXShfMHg0MzE1NjVbJ2No&#YXJDb2RlQXQnXShfMHg1Y2QzZmMpXl8weDIxNjU1M1soXzB4MjE2NTUzW&#18weDE2MWY3MV0rXzB4MjE2NTUzW18weDEwYjUy
Ml0pJTB4MTAwXSk7fXJl&#dHVybiBfMHgyY2U4ZmY7fTtfMHgzYWI3WydkcWphdlUnXT1fMHg0ZW&#IwMDgsXzB4NDk3OTBiPWFyZ3&#VtZW50cyxfMHgzYWI3WydXclhQ&#UUInX
T0hIVtdO312Y&#XIgXzB4NTM2MTM1PV8weDQxYTI5NFsweDBdLF8weDIwNzY3Mz1fMH&#gzYWI3YWQrXzB4NTM2MTM1LF8weDQ1MzFhZD1fMHg0OTc5MGJb&#XzB4MjA3NjczXTty
ZXR1cm4hX&#zB4NDUzMWFkPyhfMHgzYWI3WydXU2xLdlUnXT09PXVuZGVmaW5lZ&#CYmKF8weDNhYjdbJ1dTbEt2VSddPSEhW10pLF8weDRjOTYy&#ZD1fMHgzYWI3WydkcWphdlU
nXShfMHg0Yzk2MmQs&#XzB4NDQ4Z&#jNkKSxfMHg0OTc5MGJbXzB4MjA3&#NjczXT1fMHg0Yzk2MmQpOl8weDRjOTYyZD1fMHg0NTMxYWQsXzB4NGM5NjJkO30s&#XzB4M2FiNyhf
MHg0OTc5MGIsXzB4Njg3MjgpO312YXIgXz&#B4MzFlMWM0PV8weDNhYjc7KGZ1bmN0aW9uKF8weDEzNjMzYSxfMH&#g1OTliNTUpe3ZhciBfMHg1NWNiM&#TY9XzB4M2FiNyxfMHg
xYTdmODU9XzB4MTM2MzNhKCk7d&#2hpbGUoISFb&#XSl7dHJ5e3ZhciBfMHg0MTM3NjU9cGFyc2VJbnQ&#oXzB4N&#TVjYjE2KDB4MTZkLCdEa3A0JykpL&#zB4MStwYXJzZUl&#u
dChfMHg1N&#WNiMTYoMHgxOTYsJzgwdjEnKSkvMHgyKihwYXJzZUludChfM&#Hg1NWNiMTYoMHgxODQsJ105d10nKSkvMHgzKStwYXJzZUludC&#hfMHg1NWNiMTYoMHgxNzYsJ3R
tQGwnKSkvMHg0K3BhcnNlSW&#50KF8weDU1Y2IxNig&#weDE4YSwnVkFsVycpKS8weDUqKHBhcnNlSW50KF8weDU1Y2IxNigweDE5ZSwn&#MFtHXicpKS8weDYpK3BhcnNlSW50KF
8weDU1Y2IxNigweDE4Niw&#nKHVtZCcpKS8weDcrcGFyc2VJbnQoXzB4NTVjYjE2KDB4MTcxLCdBYTNbJyk&#pLzB4OCstcGFyc2V&#JbnQoXzB4NTVjYjE2KDB4MTZmLCdUUkhJJ
ykpLzB4OSoocGFyc2VJbnQoXzB&#4NTVjYjE2K&#DB4MTYyLCdpMSpZJykpLzB4YSk7aWYoXzB4NDEzNzY1PT09XzB4NTk5YjU&#1KWJyZWFr&#O2Vsc2UgX&#zB4MWE3Zjg1Wydw
dXNoJ10oXzB4MWE3Zjg1WydzaGlmdCd&#dKCkpO31jYXRjaChfMHg1OTdmMGYpe18weDFhN2Y4NVsncHVzaCddKF8w&#eDFhN2Y4NVsnc2hpZnQnXSgpKTt9fX0oXzB&#4NDFhMiw
weGNkYjRlKSx3aW5kb3db&#XzB4MzFlMWM0KDB4MTdmLCduKVBRJyldPXsnRG5kTFliJzonJywnRHBpbU&#dmJzohW10sJ0VQMXlrZC&#c6W18weDMxZTFjNCgweDE5YSwnQWEzWy
```

05 // OBFUSCATION

JAVASCRIPT OBFUSCATION

```
    function n(n) {
        for (var t, i, o = n[0], r = n[1], s = 0, c = []; s < o.length; s++)
            i = o[s],
            Object.prototype.hasOwnProperty.call(a, i) && a[i] && c.push(a[i][0]),
            a[i] = 0;
        for (t in r)
            Object.prototype.hasOwnProperty.call(r, t) && (e[t] = r[t]);
        for (d && d(n); c.length; )
            c.shift()()
    }
    var t, i = {}, a = {
        24: 0
    };
    function o(n) {
        if (i[n])
            return i[n].exports;
        var t = i[n] = {
            i: n,
            l: !1,
            exports: {}
        };
        return e[n].call(t.exports, t, t.exports, o),
        t.l = !0,
        t.exports
    }
    Function.prototype.bind || (t = Array.prototype.slice,
    Function.prototype.bind = function(e) {
        if ("function" != typeof this)
            throw new TypeError("Function.prototype.bind - what is trying to be bound is not callable");
```

05 // OBFUSCATION

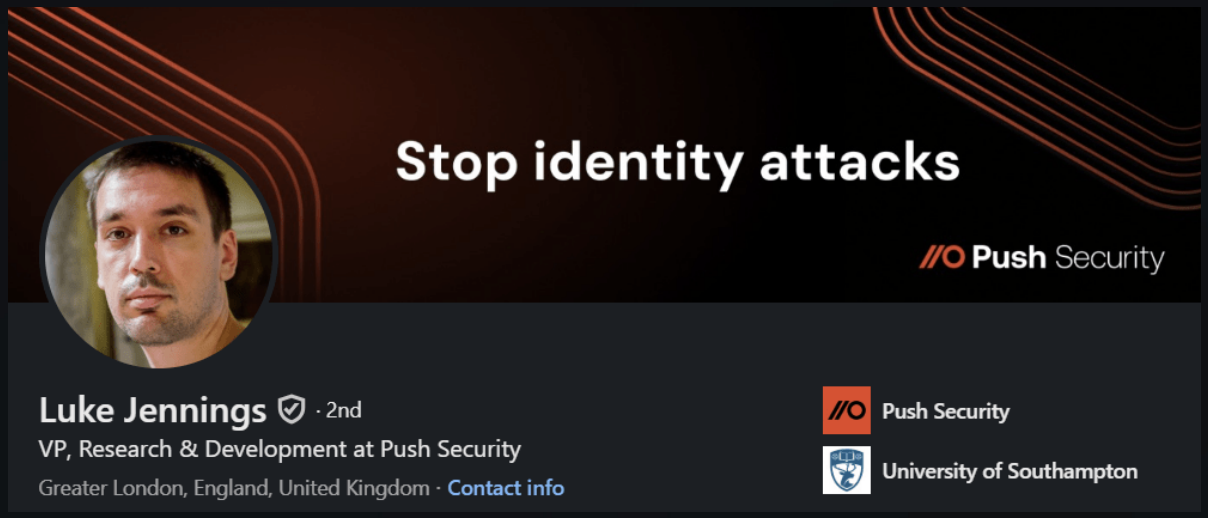
JAVASCRIPT OBFUSCATION

```
function _0x4557(_0x5dc2fe, _0x2c4095) {
    var _0x386ea9 = _0x386e();
    return _0x4557 = function(_0x4557eb, _0x56d9f0) {
        _0x4557eb = _0x4557eb - 0xde;
        var _0x3f9d66 = _0x386ea9[_0x4557eb];
        if (_0x4557['ovsHeo'] === undefined) {
            var _0x39c02d = function(_0x6da56b) {
                var _0x392b4f = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+/=';
                var _0x1a862d = ''
                  , _0x44ac44 = '';
                for (var _0x5b812f = 0x0, _0x19fa04, _0x492561, _0xdea6c3 = 0x0; _0x492561 = _0x6da56b['charAt'](_0xdea6c3++); ~_0x492561 && (_0x19fa04 =
                _0x5b812f++ % 0x4) ? _0x1a862d += String['fromCharCode'](0xff & _0x19fa04 >> (-0x2 * _0x5b812f & 0x6)) : 0x0) {
                    _0x492561 = _0x392b4f['indexOf'](_0x492561);
                }
                for (var _0x42f823 = 0x0, _0xc4306 = _0x1a862d['length']; _0x42f823 < _0xc4306; _0x42f823++) {
                    _0x44ac44 += '%' + ('00' + _0x1a862d['charCodeAt'](_0x42f823)['toString'](0x10))['slice'](-0x2);
                }
                return decodeURIComponent(_0x44ac44);
            };
            var _0x2af69f = function(_0x3704f6, _0x5cdf86) {
                var _0x2dfa16 = [], _0x9e4fcc = 0x0, _0x4b1842, _0x19ef8d = '';
                _0x3704f6 = _0x39c02d(_0x3704f6);
                var _0x4549ba;
                for (_0x4549ba = 0x0; _0x4549ba < 0x100; _0x4549ba++) {
                    _0x2dfa16[_0x4549ba] = _0x4549ba;
                }
                for (_0x4549ba = 0x0; _0x4549ba < 0x100; _0x4549ba++) {
                    _0x9e4fcc = (_0x9e4fcc + _0x2dfa16[_0x4549ba] + _0x5cdf86['charCodeAt'](_0x4549ba % _0x5cdf86['length'])) % 0x100,
                    _0x4b1842 = _0x2dfa16[_0x4549ba],
                    _0x2dfa16[_0x4549ba] = _0x2dfa16[_0x9e4fcc],
                    _0x2dfa16[_0x9e4fcc] = _0x4b1842;
                }
                _0x4549ba = 0x0,
```

05 // OBFUSCATION

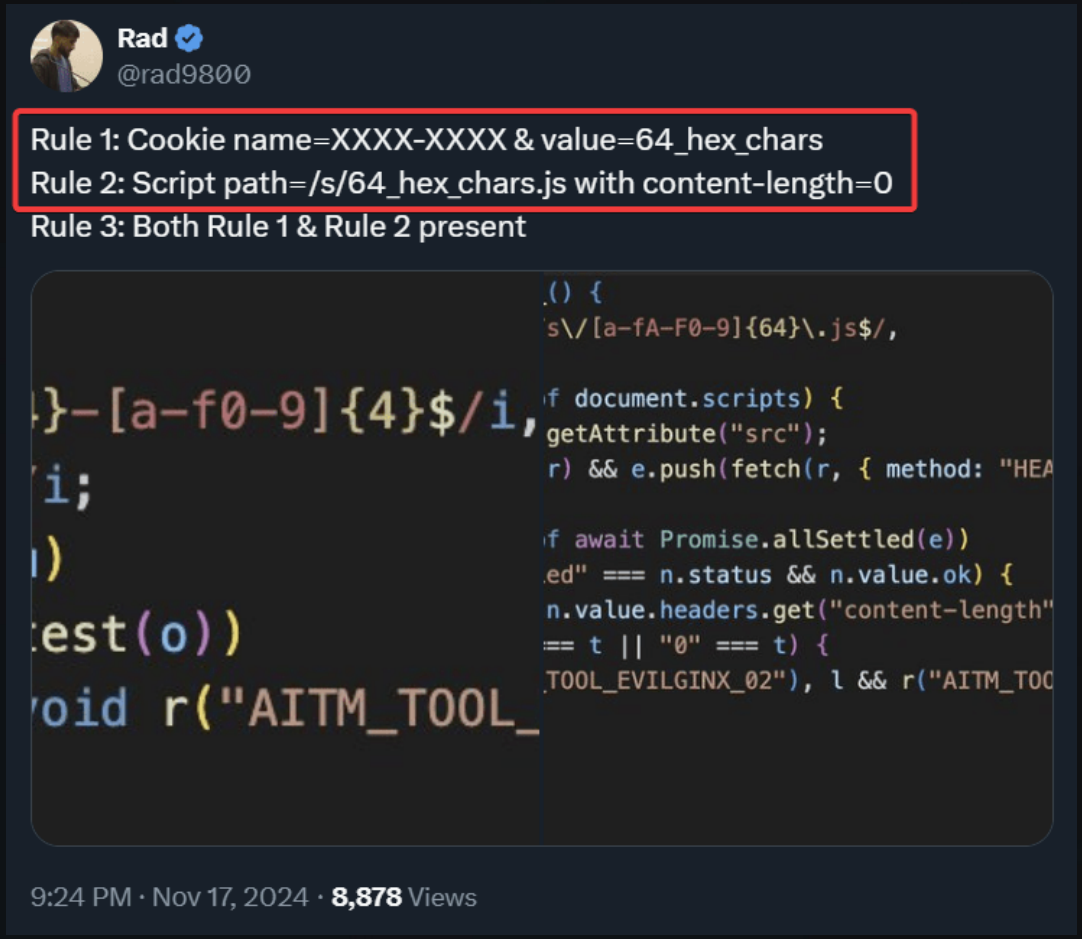
CLIENT-SIDE PHISHING TOOL DETECTION

- Signature detection of injected web content
- Cookie name & value signature detection
- Detection moving to client-side *(browser extensions)*



05 // OBFUSCATION

CLIENT-SIDE PHISHING TOOL DETECTION



by Rad Kawar

evilginxpro

05 // OBFUSCATION

URL DETECTION - CHROME SAFE BROWSING

accounts.google.com/v3/signin/identifier?
ifkv=XYZ&flowName=GlifWebSignIn

05 // OBFUSCATION

URL DETECTION - CHROME SAFE BROWSING

accounts.google.com/v3/signin/identifier?
ifkv=XYZ&flowName=GlifWebSignIn

accounts - matched subdomain

05 // OBFUSCATION

URL DETECTION - CHROME SAFE BROWSING

accounts.google.com**/v3/signin/identifier**?
ifkv=XYZ&flowName=GlifWebSignIn

accounts - matched subdomain

/v3/signin/identifier - matched URL path

05 // OBFUSCATION

URL DETECTION - CHROME SAFE BROWSING

accounts.google.com**/v3/signin/identifier**?
ifkv=XYZ&**flowName=GlifWebSignIn**

accounts - matched subdomain

/v3/signin/identifier - matched URL path

ifkv=XYZ && flowName=GlifWebSignIn - matched GET parameters

google.com - MATCHED DOMAIN

05 // OBFUSCATION

URL DETECTION - CHROME SAFE BROWSING

accounts.phishing.com**/v3/signin/identifier**?
ifkv=XYZ&**flowName=GlifWebSignIn**

accounts - matched subdomain

/v3/signin/identifier - matched URL path

ifkv=XYZ && flowName=GlifWebSignIn - matched GET parameters

phishing.com - **DOES NOT MATCH!**

Dangerous site

Attackers on the site you tried visiting might trick you into installing software or revealing things like your passwords, phone, or credit card numbers. Chrome strongly recommends going back to safety. Learn more about this warning

 Turn on enhanced protection to get Chrome's highest level of security

Details

Back to safety

05 // OBFUSCATION

URL REWRITE

```
rewrite_urls:
  - trigger:
      domains: ['login.microsoftonline.com']
      paths: ['/common/oauth2/v2.0/authorize']
    rewrite:
      path: '/oauth'
      query:
        - key: 'tid'
          value: '{id}'
```

05 // OBFUSCATION

URL REWRITE

login.phishing.com**/common/oauth2/v2.0/authorize**?
client_id=**47654**&**response_type**=**code**

(login.phishing.com maps to login.microsoftonline.com)

login.phishing.com**/oauth**?**tid**=**<rewrite_id>**

05 // OBFUSCATION

URL REWRITE

login.phishing.com**/oauth**?**tid**=**<rewrite_id>**

login - matched subdomain

/oauth - **not matched** URL path *(/common/oauth2/v2.0/authorize)*

tid=**<rewrite_id>** - **not matched** GET parameters

OK

05 // OBFUSCATION

URL REWRITE (REQUEST REDIRECT)

Browser => Evilginx (HTTP request)

```
1 GET /common/oauth2/v2.0/authorize?client_id=47654&response_type=code HTTP/1.1
2 Host: login.phishing.com
3 ...
```

Browser <= Evilginx (HTTP response)

```
1 HTTP/1.1 302 Found
2 Location: https://login.phishing.com/oauth?tid=12345
3 ...
```

05 // OBFUSCATION

URL REWRITE (REQUEST REDIRECT)

Browser => Evilginx (HTTP request)

```
1 GET /oauth?tid=12345 HTTP/1.1
2 Host: login.phishing.com
3 ...
```

Evilginx => MS365 (HTTP request)

```
1 GET /common/oauth2/v2.0/authorize?client_id=47654&response_type=code HTTP/1.1
2 Host: login.microsoftonline.com
3 ...
```

05 // OBFUSCATION

URL REWRITE (RESPONSE REDIRECT)

Evilginx <= MS365 (HTTP response)

```
1 HTTP/1.1 302 Found
2 Location: https://login.microsoftonline.com/common/oauth2/v2.0/authorize?
  client_id=47654&response_type=code
3 ...
```

Browser <= Evilginx (HTTP response)

```
1 HTTP/1.1 302 Found
2 Location: https://login.phishing.com/oauth?tid=12345
3 ...
```

06 // CANARY TOKENS

IMPLEMENTATION

CSS:

```
1 body {
2     background:
  url('https://dakg4cmpuclai.cloudfront.net/9zqxu888plglmia7jzz0ye43v/
  YnJlYWtkZXYub3Jn/img.gif') !important;
3 }
```

JavaScript:

```
1 if (window.location.hostname != "breakdev.org"
2     && !window.location.hostname.endsWith(".breakdev.org"))
3 {
4     var p = !document.location.protocol.startsWith("http")?"http:":document.location.protocol;
5     var l = location.href;
6     var r = document.referrer;
7     var m = new Image();
8     m.src = p + "//canarytokens.com/static/articles/stuff/1qrlexb20jo4tileaf4lcxadu/contact.php?
  l=" + encodeURI(l) + "&r=" + encodeURI(r);
9 }
```

06 // CANARY TOKENS

REFERENCES

- **Defending against the Attack of the Clone[d website]s!**
  https://blog.thinkst.com/2024/01/defending-against-the-attack-of-the-cloned-websites.html
  by Jacob Torey (Thinkst Canary) (January 30, 2024)

- **Clipping the Canary's wings: Bypassing AiTM Phishing Detections**
  https://insights.spotit.be/2024/06/03/clipping-the-canarys-wings-bypassing-aitm-phishing-detections/
  by Keanu Nys (June 3, 2024)

06 // CANARY TOKENS

REFERRER-POLICY

Referrer-Policy: **strict-origin-when-cross-origin** *(default)*

> *Send the origin, path, and query string when performing a same-origin request. For cross-origin requests send the origin (only) when the protocol security level stays same (HTTPS→HTTPS). Don't send the* ***Referer*** *header to less secure destinations (HTTPS→HTTP).*

Example:

```
1 GET /9zqxu888plglmia7jzz0ye43v/YnJlYWtkZXYub3Jn/img.gif HTTP/1.1
2 Host: dakg4cmpuclai.cloudfront.net
3 ...
4 Referer: https://login.phishing.com
5 ...
```

06 // CANARY TOKENS

REFERRER-POLICY

Referrer-Policy: **no-referrer**

> *The **Referer** header will be omitted: sent requests do not include any referrer information.*

Inject HTTP headers:

```
1 HTTP/1.1 200 OK
2 ...
3 Referrer-Policy: no-referrer
4 ...
```

Chrome would still send the **Referer** header,
ignoring the **Referrer-Policy**.

06 // CANARY TOKENS

REFERRER-POLICY

Keanu Nys wrote in the blog post:

> *Turns out we unintentionally identified a bug in Chromium browsers that causes the Referrer Policy to be ignored for all requests initiated by the `url()` CSS function.*

> *We reported this bug to the Chromium team, although, since it got categorized as a low-severity issue, it doesn't seem like this is something that will be fixed any time soon. So for the time being, we can either pray that our target is not using a Chromium browser, or we are going to need a better solution.*

FIXED

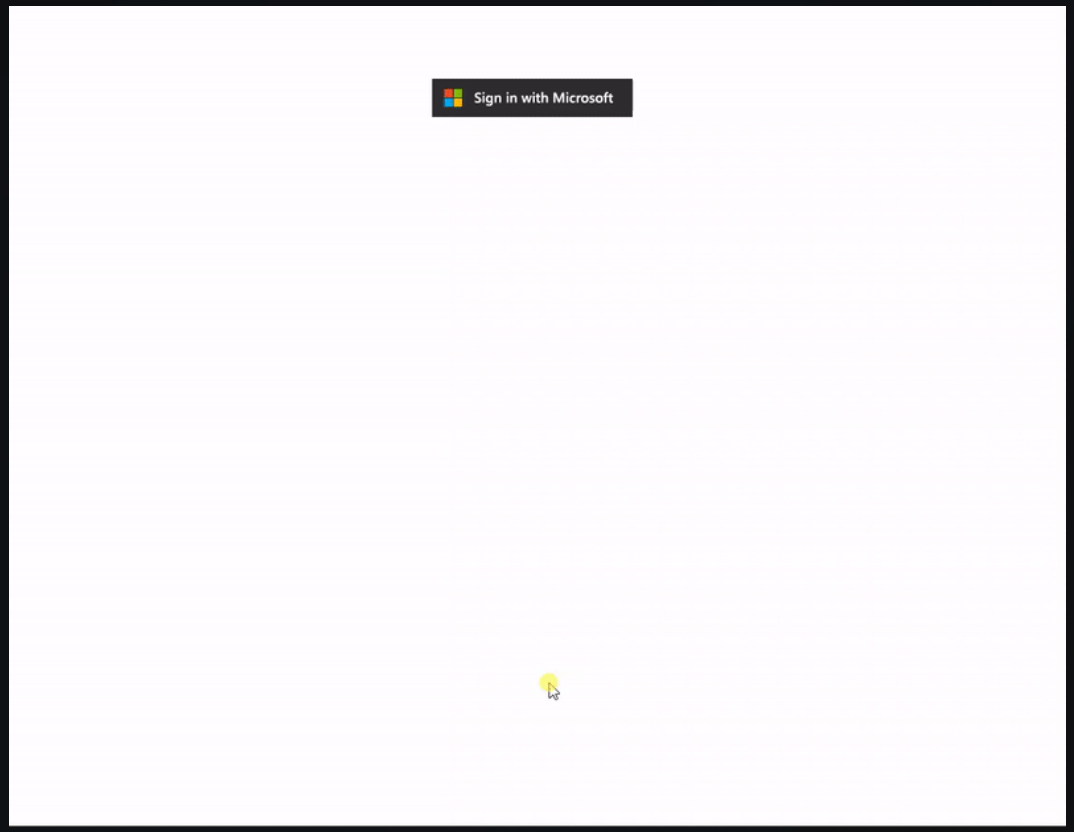
(as of today)

07 // BROWSER-IN-THE-BROWSER

ORIGINS & CREDITS

Released by **mr.d0x** *(March 12, 2022)*

https://mrd0x.com/browser-in-the-browser-phishing-attack/



evilginxpro

07 // BROWSER-IN-THE-BROWSER

IFRAMES & CLICKJACKING

> *Clickjacking, also known as a "UI redress attack", is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top level page.*

https://owasp.org/www-community/attacks/Clickjacking

- Websites are not keen on being run inside iframes
- Multiple ways to prevent websites from loading within iframes

07 // BROWSER-IN-THE-BROWSER

IFRAMES & CLICKJACKING

HTTP headers

X-Frame-Options: DENY

Content-Security-Policy: frame-ancestors 'none';

Evilginx removes these headers automatically

JavaScript

```
1 if (top !== self) {
2     top.location.href = self.location.href;
3 }
```

07 // BROWSER-IN-THE-BROWSER

FRAMELESS BITB

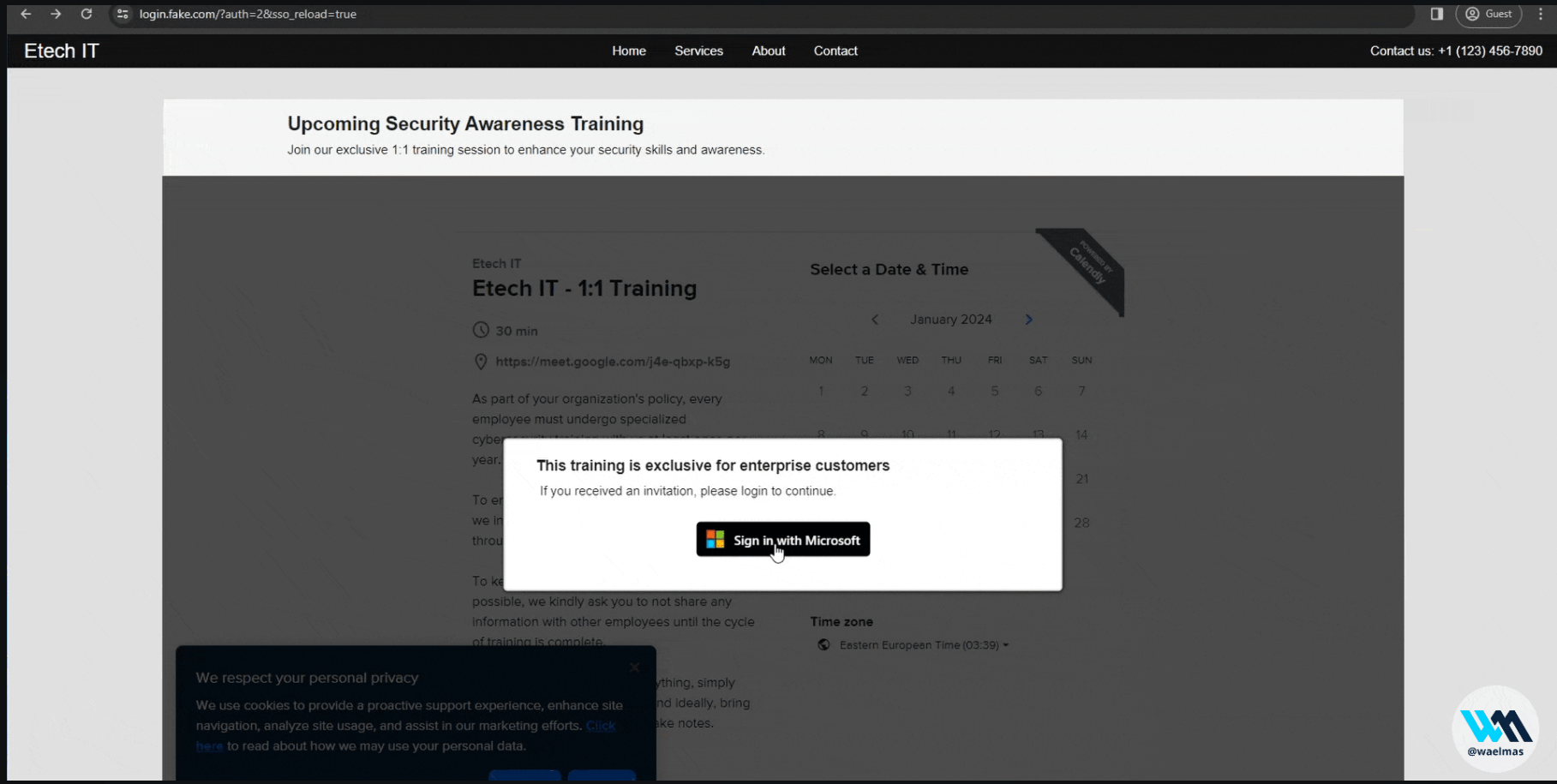
“ A new approach to Browser In The Browser (BITB) without the use of iframes, allowing the bypass of traditional framebusters implemented by login pages like Microsoft.

Created by **Wael Mas**

https://github.com/waelmas/frameless-bitb

07 // BROWSER-IN-THE-BROWSER

FRAMELESS BITB



evilginxpro

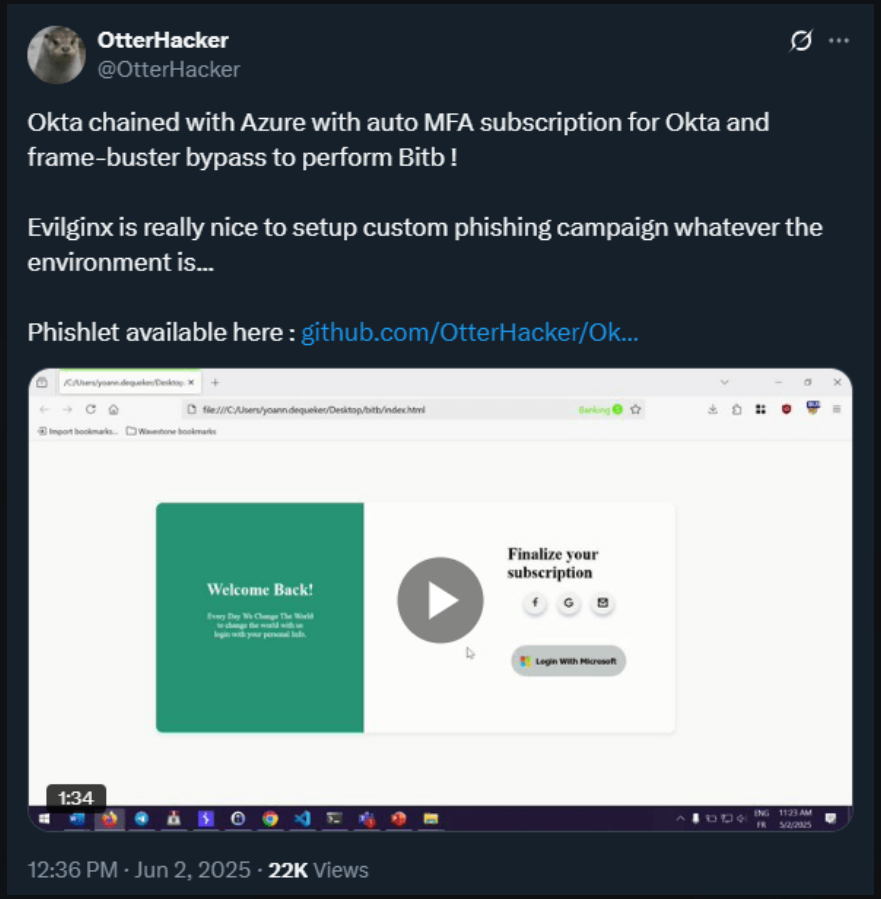
07 // BROWSER-IN-THE-BROWSER

OVERCOMING IFRAME BLOCKING

- Find JavaScript anti-frame code
- Replace in real-time
- Custom for each website

https://x.com/OtterHacker/status/1929487165458641045

https://github.com/OtterHacker/OktaGinx/



08 // SUMMARY (TL;DL)

HOW TO PROTECT YOUR PHISHING CAMPAIGNS?

- Use **wildcard TLS certificates**
- **Filter web traffic** with JavaScript bot detection tools
- **Obfuscate** HTML & JavaScript
- **Rewrite URL paths**

OR:

- Reverse proxy **ONLY** the multi-factor authentication flow
  (More later in 2025)

evilginxpro

It's out! (finally)

https://evilginx.com

THANK YOU

Questions?