THOMAS MORE

Automated Red Team Setup

Realisatiedocument

ThomasDeboel
Student Bachelor in Elektronica-ICT – Cloud & Cyber Security

2024 - 2025

Inhoudsopgave

1. Inleiding

Binnen het domein van cybersecurity is het uitvoeren van red team operaties een essentieel onderdeel geworden van moderne beveiligingstrategieën. Deze operaties simuleren realistische aanvallen op IT-omgevingen om kwetsbaarheden aan het licht te brengen voordat anderen dat doen. De Technische complexiteit en manuele opzet van deze infrastructuren zorgen echter voor struikelblokken voor efficiëntie en schaalbaarheid.

Het stageproject Red Team Infrastructure biedt hiervoor een oplossing. Het betreft een geautomatiseerde, modulaire en reproduceerbare red team-omgeving die volledig wordt uitgerold op Hetzner Cloud met behulp van Infrastructure as Code, gerealiseerd via OpenTofu. De omgeving is ontworpen voor tijdelijke inzet, waarbij snelle provisioning mogelijk is en er een duidelijke scheiding bestaat tussen verschillende klanten of engagements.

De uitrol, configuratie en afbraak van de infrastructuur worden aangestuurd door een Python-wrapper rond OpenTofu. Dankzij dit automatiseringsmechanisme kunnen tot 255 afzonderlijke omgevingen parallel worden beheerd, zonder handmatige tussenkomst.

In dit realisatiedocument wordt de volledige implementatie van het project besproken. Hoofdstuk 2 behandelt de analyse van de gebruikte technologieën en de onderliggende keuzes. Hoofdstuk 3 gaat in op de technische werking, architectuur en de afzonderlijke componenten van de omgeving. Het document sluit af met een besluit en een bijlage met ondersteunende informatie.

2. Analyse

2.1. Phishing

De onderstaande tabel maakt gebruik van de Weighted Ranking Method (WRM) en vergelijkt verschillende e-mailoplossingen op basis van meerdere beoordelingscriteria. De focus ligt hierbij op:

- E-mailbezorging & spoofing: Hoe gemakkelijk is het om e-mails te versturen, en in hoeverre is het mogelijk om de afzender te vervalsen (spoofing)?
- Mailopvolging: Hoe eenvoudig is het om de inhoud van een verstuurde e-mail te manipuleren of te volgen, bijvoorbeeld voor phishingcampagnes?
- Uitbreidbaarheid: In welke mate kan de oplossing geïntegreerd worden met andere tools of services (bijv. API-koppelingen)?
- Kosten: Is de service gratis beschikbaar of is er een betaalmodel van toepassing?

Op basis van deze criteria worden de oplossingen gewogen en beoordeeld. Dit biedt een objectieve vergelijking bij de keuze voor een geschikte oplossing binnen het project.

	gophish	SET	Metasploit Pro	HiddenEye
Emailbezorging /spoofing	5	3	4	2
Opvolging	5	3	4	1
Aanpasbaarheid	4	5	3	1
Uitbreidbaarheid	4	5	4	1
Prijs	5	5	1	5

Gophish is een van de meest populaire open-source phishing frameworks. Het biedt een duidelijke handleiding voor het opzetten van gepersonaliseerde phishing campagnes. Met de web interface en trackingmogelijkheiden en volledige controle over de e-mailtemplates en integratie met Evilginx. Daarnaast is er goede documentatie dat ervoor zorgt dat GoPhish eenvoudig inzetbaar is binnen zowel kleine als grootschalige scenario's.(Wright, z.d.)

SET (Social Engineering Toolkit) is een framework voor social engineering-aanvallen. Het ondersteund onder andere spearphishing, USB-attacks en web cloning. Hoewel SET veelzijdig is vereist het veel handmatige interactie en is het minder geschikt voor grootschalige en geautomatiseerde campagnes.(Trustedsec, z.d.)

Metasploit Pro is een commerciele extensie van de opensource Metasploit framework. Het bevat wel modules voor social engineering en phishing maar is meer gericht op exploitatie en penetratietesten. De prijs en de beperkte focus op phishing maakt het niet geschikt voor het project.(*Docs @ Rapid7*, z.d.)

HiddenEye is een eenvoudige tool voor credential harvesting via valse loginpagina's. Hoewel handig voor snelle demonstraties of labtesting, is het project inmiddels verouderd en wordt het niet meer onderhouden. De tool mist trackingmogelijkheden, schaalbaarheid en betrouwbare e-mailfunctionaliteit.(Darkmidus, z.d.)

We hebben gekozen om GoPhish te gebruiken met combinatie van Evilginx voor credential harvest. Deze combinatie maakt het mogelijk om een zeer goede phishing website te combineren met phishing mails.

2.2. C2 server

Voor de evaluatie van de Command and Control (C2) servers wordt eveneens gebruikgemaakt van de Weighted Ranking Method (WRM). Hierbij worden de volgende criteria gehanteerd:

- Gebruiksvriendelijkheid: Hoe intuïtief en efficiënt is de interface voor operators? Hoe snel kan een gebruiker aan de slag zonder uitgebreide voorkennis?
- Functionaliteiten: Welke ingebouwde features biedt de C2-server, zoals op afstand uitvoeren van commando's, file transfers, persistence-mechanismen, encryptie, enzovoort?
- Beschikbare payloads: Welke payloads worden ondersteund en in welke formaten/platformen (bijvoorbeeld Windows, Linux, macOS)? Zijn deze makkelijk aan te passen of te genereren?
- Kosten: Is de tool volledig gratis, open-source of vereist het een (commerciële) licentie?
- Documentatie & community: Is er voldoende technische documentatie beschikbaar? Zijn er actieve communities, tutorials of voorbeelden die het gebruik ondersteunen?

Door deze criteria te combineren in een gewogen score, kan op objectieve wijze worden bepaald welke C2-server het best aansluit bij de behoeften van dit project.

	ninjaC2	phosC2	Covenant	Caldera	Metasploit	Infection monkey	Mythic C2	sliver
Gebruiksgemak	4	3	4	4	4	4	4	4
Functionaliteit	3	3	3	4	4	3	5	5
Payload support	3	3	3	3	4	3	5	5
Prijs	5	5	5	5	3	5	5	5
Documentatie	3	2	4	4	5	4	5	4

NinjaC2 is een relatief nieuwe en lichtgewicht command-and-control framework. Het biedt basisfunctionaliteiten en is eenvoudig te gebruiken, maar mist nog geavanceerde evasietechnieken en brede ondersteuning voor verschillende platformen. De documentatie is beperkt, waardoor het vooral geschikt is voor kleine tot middelgrote testomgevingen.(Ahmedkhlief, z.d.)

PhosC2 is een open-source framework dat zich richt op gebruiksvriendelijkheid en eenvoud. Hoewel het een solide basis biedt, is het qua functionaliteit en payload-ondersteuning minder uitgebreid dan andere opties. Het is gratis en toegankelijk, maar mist geavanceerde features en een uitgebreide community.(*Overview — PoshC2*, z.d.)

Covenant is een populaire .NET-gebaseerde C2-server met een gebruiksvriendelijke webinterface. Het ondersteunt verschillende platforms en biedt goede modulariteit. De tool is open-source en relatief actief,

maar loopt soms achter op de nieuwste evasietechnieken en heeft beperkte documentatie in vergelijking met grotere projecten.(Cobbr, z.d.)

Caldera, ontwikkeld door MITRE, is een geavanceerd open-source platform gericht op automatische aanvalsemulatie. Het combineert een breed scala aan technieken met AI-ondersteuning om realistische scenario's te creëren. Het is zeer krachtig voor onderzoek en simulaties, maar minder geschikt als standaard C2-server door de complexiteit en focus op automatisering.(*Welcome to MITRE Caldera's documentation! — caldera documentation*, z.d.)

Metasploit is een gevestigde naam in de securitywereld met uitgebreide exploit- en payloadmogelijkheden. De Pro-versie bevat een C2-module, maar deze is niet zo gespecialiseerd of flexibel als dedicated C2-frameworks. De tool is commercieel, wat een hogere kost betekent, maar heeft uitstekende documentatie en community-ondersteuning.(*Docs @ Rapid7*, z.d.)

Infection Monkey is een open-source breach en aanvalssimulatieplatform dat zich richt op het testen van netwerkweerstand. Het bevat een eenvoudige C2-functionaliteit, maar is minder gefocust op stealth of uitgebreide payloadmogelijkheden. Het is vooral nuttig voor netwerkpenetratietests, niet als volledig C2-framework.(*Infection Monkey | Akamai*, z.d.)

Mythic is een modern, open-source C2-framework dat bekend staat om zijn uitgebreide functionaliteit, gebruiksvriendelijke interface en brede ondersteuning voor verschillende platformen. Het ondersteunt geavanceerde evasietechnieken, modulaire payloads en biedt sterke documentatie en community-ondersteuning. Mythic wordt actief onderhouden en is daarom een uitstekende keuze voor zowel red teaming als securityonderzoek.(*Mythic | Mythic Documentation*, z.d.)

Sliver is een krachtig en flexibel open-source C2-framework met ondersteuning voor meerdere platforms en geavanceerde evasietechnieken. Het heeft een actieve community en goede documentatie, en wordt vaak gezien als een sterke concurrent van Mythic. Het is lichtgewicht en snel, met een focus op stealth en modulariteit.(*Sliver Docs: Getting Started*, z.d.)

Wij hebben gekozen voor Mythic. Dit framework biedt de beste balans tussen gebruiksvriendelijkheid, functionaliteit en stealth-mogelijkheden. De brede ondersteuning voor verschillende platformen en modulaire architectuur maken Mythic ideaal voor complexe red team-operaties. Daarnaast zorgt de actieve ontwikkeling en uitgebreide documentatie ervoor dat wij altijd up-to-date blijven met de nieuwste aanvalstechnieken en verdedigingen.

2.3. Attack Server

In de volgende tabel worden verschillende attack servers met elkaar vergeleken aan de hand van de Weighted Ranking Method (WRM). Hierbij wordt gekeken naar de volgende beoordelingscriteria:

- Toolset: Hoe uitgebreid is het standaardpakket aan voorgeïnstalleerde tools (zoals scanners, exploit frameworks, pivotingtools, enzovoort)?
- Gebruiksvriendelijkheid: Hoe eenvoudig is de attack server in gebruik voor een operator? Hoe snel kan men effectief aan de slag?
- Cloud-deployability: Hoe gemakkelijk is het om de omgeving automatisch of handmatig te implementeren in een cloudomgeving (zoals Hetzner, AWS, Azure)?
- Uitbreidbaarheid: Hoe eenvoudig is het om extra tools of scripts toe te voegen aan het bestaande platform?
- Logging en monitoring: In welke mate is het mogelijk om logging te implementeren of te integreren met externe loggingtools (zoals Grafana, Loki, Elastic of Alloy)?

Deze criteria worden gewogen en vertaald naar een totaalscore per oplossing, zodat de meest geschikte attack server voor dit project geselecteerd kan worden.

	Kali	Exegol	Blackarch	ParrotOS
Aantal tools	4	3	5	4
Gebruiksgemak	4	5	3	4
Cloud compatability	3	5	2	2
Extra tools	4	4	4	4
logging	3	5	2	3

Kali Linux is een op Debian gebaseerde distributie, specifiek ontworpen voor penetratietesten en digital forensics. Het beschikt over een gebruiksvriendelijke desktopomgeving met een GUI en CLI-interface, en is beschikbaar als ISO, VM-image en container. Hoewel Kali initieel niet ontworpen is voor de cloud, kan het relatief eenvoudig worden uitgerold als virtuele machine op Cloud platforms zoals AWS, Azure of Hetzner. De voorgeïnstalleerde selectie van ongeveer 600 tools dekt een breed spectrum aan offensieve technieken. Extra tools kunnen eenvoudig worden toegevoegd via apt, pip of manuele installatie, en upgrades verlopen via het standaard APT-systeem. Logging is niet standaard geconfigureerd voor red team operaties, maar kan handmatig worden ingesteld via syslog of externe forwarding.
(*Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution*, 2025)

BlackArch is een op Arch Linux gebaseerde distributie die zich profileert als een ultracomplete pentestomgeving, met meer dan 2600 tools in de repository. De interface is primair CLI-gebaseerd en vereist een zekere mate van expertise om efficiënt te gebruiken. De installatie is relatief complex en niet standaard voorzien van een desktopomgeving. BlackArch is minder Cloud vriendelijk: het opzetten van een stabiele VM is tijdrovend, en automatisatie van deployments vereist extra scripting. Hoewel de hoeveelheid tools indrukwekkend is, is de praktische bruikbaarheid lager door beperkte documentatie en gebrek aan consistentie. Het toevoegen of updaten van tools gebeurt via pacman, het package managementsysteem van Arch, maar conflicten en dependency-issues zijn niet ongewoon. Logging is minimaal aanwezig en vereist manuele configuratie.(*BlackArch Linux - Penetration Testing Distribution*, z.d.)

Parrot OS is, net als Kali, gebaseerd op Debian en mikt op zowel privacy bewuste gebruikers als pentesters. De distributie heeft een moderne GUI en een brede set tools voor offensieve en defensieve securitytaken. Clouddeployment is mogelijk via vooraf geconfigureerde images, maar vereist bijkomende setup voor automatisatie. Het gebruik van tools is vergelijkbaar met Kali, en extra software kan worden toegevoegd via APT of third-party scripts. Loggingmogelijkheden zijn vergelijkbaar met Kali Linux: niet standaard gericht op red teaming, maar aanpasbaar via klassieke Linux-methodes.(*Parrot Security*, z.d.)

Exegol daarentegen is geen volledige OS-distributie, maar een container gebaseerde aanvalsomgeving, gebouwd bovenop Docker. Het wordt geleverd met een command line interface, maar biedt daarnaast een optionele web interface via noVNC, waarmee gebruikers een desktopomgeving kunnen openen in de browser. Dit maakt Exegol uitermate geschikt voor cloudgebruik: de containers kunnen snel opgestart, gekloond en afgebroken worden, en passen perfect in geautomatiseerde pipelines of tijdelijke red team engagementen. Tools binnen Exegol zijn gericht op offensive operations en kunnen eenvoudig worden toegevoegd of aangepast via de Dockerfile of door extra installaties in de actieve container. Het loggen van acties kan op meerdere niveaus worden ingeschakeld, waaronder Docker logging drivers, auditlogs of forwarding naar centrale SIEM-systemen. Deze flexibiliteit en modulariteit maken Exegol ideaal voor omgevingen waar snelheid, herhaalbaarheid en schaalbaarheid centraal staan.
We hebben gekozen voor Exegol voor deze redenen en door dat het al intern gebruikt wordt wat de overhandiging vergemakkelijkt. (*Exegol: professional hacking setup — Exegol documentation*, z.d.)

2.4. Cloud

Voor de vergelijking van verschillende cloudproviders wordt opnieuw gebruikgemaakt van de Weighted Ranking Method (WRM). De volgende criteria worden hierbij in overweging genomen:

- Prijs: De maandelijkse of uurprijs voor het draaien van een standaard serverconfiguratie (inclusief opslag en bandbreedte).
- Beschikbaarheid: Hoe eenvoudig en snel is het om een server in de gewenste regio of locatie te verkrijgen, rekening houdend met eventuele beperkingen of uitputting van capaciteit?
- Datacenterlocaties: Waar bevinden de datacenters zich geografisch? Dit speelt een rol in juridische aspecten en operationele dekking.
- Automatiseerbaarheid: In welke mate is de cloudomgeving geschikt voor automatisering via tools zoals Terraform, Ansible, API's of eigen scripts?

Op basis van deze criteria worden de cloudproviders objectief vergeleken en gewogen, zodat er een weloverwogen keuze gemaakt kan worden voor de infrastructuur van dit project.

	AWS	Azure	Hetzner	Google Cloud
Prijs	2	2	4	2
Verkrijgbaarheid	5	5	4	5
Plaats	5	5	5	5
Automatisering	5	5	3	5

AWS (**Amazon Web Services**) biedt een pay-as-you-go model, wat voordelig kan zijn, maar het kan ook duur uitvallen bij intensief gebruik. De beschikbaarheid is hoog, met meerdere datacenters wereldwijd, en de response time is over het algemeen snel, afhankelijk van de regio. AWS heeft sterke automatiseringsmogelijkheden via tools zoals CloudFormation en Lambda.(*Cloud computing services - Amazon Web Services (AWS)*, z.d.)

Azure is competitief geprijsd en biedt verschillende prijsmodellen, inclusief kortingen voor langdurig gebruik. De beschikbaarheid is eveneens hoog, met een sterke aanwezigheid in Europa en de VS. De response time is meestal goed, afhankelijk van de gekozen services. Azure ondersteunt automatisering via Azure Resource Manager en Azure DevOps.

Google Cloud heeft een concurrerende prijsstelling met kortingen voor langdurig gebruik en biedt een hoge beschikbaarheid met wereldwijde dekking. De response time is over het algemeen snel, en de automatiseringsmogelijkheden zijn sterk, met tools zoals Google Cloud Deployment Manager en Cloud Functions.(*Cloud Computing Services | Google Cloud*, z.d.)

We gekozen voor **Hetzner**. De belangrijkste reden voor deze keuze is de kosteneffectiviteit. Hetzner biedt een pay-as-you-go model met een maximumbedrag per maand, wat budgetbeheer vergemakkelijkt. Bovendien wordt Hetzner al intern gebruikt, wat de overgang naar deze Cloud omgeving vergemakkelijkt en zorgt voor een vertrouwde werkomgeving.(*Cheap dedicated servers, cloud & hosting from Germany*, z.d.)

2.5. Cloud deployment

Voor de beoordeling van verschillende cloud deployment tools wordt gekeken naar hun geschiktheid voor het automatisch uitrollen en beheren van infrastructuur binnen dit project. De tools worden vergeleken op basis van de volgende criteria:

- Gebruiksvriendelijkheid: Hoe eenvoudig is het om de tool te installeren, configureren en gebruiken? Is de leercurve geschikt voor een gemiddelde gebruiker?
- Kost: Is de tool gratis beschikbaar (open-source), of vereist deze een commerciële licentie voor volledige functionaliteit?
- Integratievermogen: In hoeverre kan de tool samenwerken met andere oplossingen zoals cloudplatforms (bijv. AWS, Azure, Hetzner), CI/CD-pijplijnen, of configuratiebeheertools (zoals Ansible)?
- Beschikbare documentatie en community: Hoe uitgebreid en toegankelijk is de officiële documentatie? Zijn er actieve communities, tutorials of voorbeelden beschikbaar die het gebruik vergemakkelijken?

Door het toepassen van de WRM op deze criteria ontstaat een gewogen vergelijking die helpt bij het selecteren van de meest geschikte deploymenttool voor deze infrastructuur.

	Terraform	OpenTofu	Chef	Jenkins	Salt	Cloud Init	Ansible
Gebruiksgemak	4	4	2	3	2	4	4
Kost	5	0	4	0	0	0	0
Samenwerking met andere tools	4	4	4	3	4	4	4
Documentatie	4	4	3	4	2	3	4

Terraform is een populaire tool voor infrastructuur als code, die een declaratieve benadering biedt voor het beheren van cloudresources. Het is gebruiksvriendelijk en heeft een sterke community, maar de kosten kunnen oplopen bij gebruik van bepaalde cloudproviders. Terraform integreert goed met andere tools en heeft uitgebreide documentatie, wat het een aantrekkelijke keuze maakt.(*Terraform | HashiCorp Developer*, z.d.)

OpenTofu is een open-source alternatief voor Terraform en biedt vergelijkbare functionaliteiten. Het is ontworpen met gebruiksgemak in gedachten en is kosteneffectief, aangezien het gratis is. De samenwerking met andere tools is vergelijkbaar met Terraform, maar de documentatie is nog in ontwikkeling, wat een uitdaging kan zijn voor nieuwe gebruikers.(*OpenTofu*, z.d.)

Chef is een krachtige configuratiebeheertool die een meer complexe leercurve heeft. De kosten kunnen hoger zijn, vooral bij gebruik van de commerciële versie. Chef biedt goede integratiemogelijkheden met andere tools, maar de documentatie kan soms overweldigend zijn voor beginners.(Software, 2021)

Ansible is eenvoudig te gebruiken en vereist geen agent-installatie, wat het aantrekkelijk maakt voor snelle implementaties. De kosten zijn laag, aangezien het open-source is. Ansible werkt goed samen met andere tools en heeft uitgebreide documentatie, wat het een populaire keuze maakt voor automatisering.(*Ansible collaborative*, z.d.)

Jenkins is een veelgebruikte tool voor continue integratie en continue levering (CI/CD). Het is gebruiksvriendelijk, maar kan complex worden naarmate de configuratie groeit. De kosten zijn afhankelijk van de gebruikte plugins en infrastructuur. Jenkins integreert goed met andere tools, en de documentatie is uitgebreid, wat de leercurve vergemakkelijkt.(*Jenkins*, z.d.)

Salt is een configuratiebeheertool die bekend staat om zijn snelheid en schaalbaarheid. Het heeft een steilere leercurve en kan hogere kosten met zich meebrengen. Salt werkt goed samen met andere tools, maar de documentatie kan soms onduidelijk zijn.(*Saltproject.io*, z.d.)

Cloud-init is een tool voor het initialiseren van cloudinstellingen en is eenvoudig te gebruiken. Het is kosteneffectief, aangezien het vaak gratis wordt aangeboden door cloudproviders. Cloud-init integreert goed met andere tools en heeft redelijke documentatie, hoewel het soms beperkt kan zijn.(*cloud-init 25.1.2 documentation*, z.d.)

In onze omgeving maken we gebruik van OpenTofu, Cloud-init en Ansible vanwege hun gebruiksvriendelijkheid, kosteneffectiviteit en sterke samenwerking met andere tools. Deze keuze stelt ons in staat om efficiënt te werken en te profiteren van de uitgebreide documentatie die deze tools bieden.

2.6. Proxy

Voor de selectie van een geschikte proxy-oplossing binnen deze infrastructuur worden verschillende tools geëvalueerd op basis van de Weighted Ranking Method (WRM). De beoordeling is gebaseerd op de volgende criteria:

- Gebruiksvriendelijkheid: Hoe eenvoudig is het om de proxy te installeren, configureren en beheren? Is het beheer intuïtief, en vereist het veel technische kennis?
- Licentiemodel: Is de tool gratis/open-source beschikbaar, of vereist deze een commerciële licentie?
- Automatiseerbaarheid: In hoeverre is de proxy-oplossing te automatiseren via scripts, API's of tools zoals Ansible of Terraform?
- Installatiegemak (setup): Hoe complex is het initiële opzetproces? Wordt er ondersteuning geboden voor standaardconfiguraties of templates?

Op basis van deze criteria wordt er een gewogen vergelijking gemaakt om de meest geschikte proxy-oplossing te bepalen voor dit project, met focus op balans tussen controle, efficiëntie en eenvoud van beheer.

	Nginx Proxy Manager	Caddy	Traefik	HAproxy	Nginx	Apache
Gebruiksgemak	4	5	3	2	3	3
Kost	0	2	2	3	3	0
Automatisatie	1	4	4	3	3	3
Setup	4	4	3	4	2	3

Nginx Proxy Manager is zeer gebruiksvriendelijk en biedt een webinterface die eenvoudige configuratie mogelijk maakt. Het is open source en heeft een actieve community, maar de automatiseringsmogelijkheden zijn beperkt. Dit maakt het een goede keuze voor gebruikers die een eenvoudige oplossing zoeken zonder veel technische complicaties.
(*Nginx Proxy Manager*, z.d.)

Caddy is ook gebruiksvriendelijk en staat bekend om zijn automatische HTTPS-configuratie. Het is open source, met een commerciële versie voor extra functies. Caddy kan eenvoudig worden geconfigureerd via configuratiebestanden en API, wat het ideaal maakt voor snelle implementaties en gebruik in dynamische omgevingen.(Server, z.d.)

Traefik Proxy is ontworpen voor dynamische configuratie en integreert naadloos met containerplatforms zoals Docker en Kubernetes. Het is open source en biedt een commerciële versie voor extra ondersteuning. Traefik is zeer geschikt voor geautomatiseerde omgevingen, omdat het automatisch services kan detecteren en configureren.(*Traefik, The Cloud Native Application Proxy | Traefik Labs*, z.d.)

HAProxy is een krachtige tool die bekend staat om zijn snelheid en betrouwbaarheid, maar het is minder gebruiksvriendelijk dan de andere opties. Het vereist meer configuratie-inzicht en kan hogere kosten met zich meebrengen bij gebruik van de commerciële versie. HAProxy kan worden geautomatiseerd met scripts

en configuratiebeheer tools, maar de setup kan complex zijn voor beginners.(*HAProxy - the reliable, high perf. TCP/HTTP load balancer*, z.d.)

Nginx is een flexibele en populaire keuze die open source is. Het vereist enige kennis van configuratiebestanden, maar biedt uitgebreide mogelijkheden voor geavanceerde configuraties. Nginx kan worden geautomatiseerd met configuratiebeheer tools, maar de setup kan uitdagend zijn voor nieuwe gebruikers.(*nginx*, z.d.)

Apache is een bekende en breed gebruikte tool die volledig open source is. Het biedt veel configuratiemogelijkheden, maar kan complex zijn afhankelijk van de gewenste functionaliteit. Apache kan ook worden geautomatiseerd met configuratiebeheer tools zoals Ansible of Puppet, maar vereist enige kennis om effectief te gebruiken.(Group, z.d.)

In onze omgeving hebben we gekozen voor Nginx, omdat we deze tool al het meest hebben gebruikt en het gemakkelijk samenwerkt met Certbot voor automatische HTTPS-configuratie. Deze combinatie biedt ons zowel de vertrouwdheid als de functionaliteit die we nodig hebben voor onze projecten.(*Certbot*, z.d.)

2.7. VPN

Voor de keuze van een geschikte VPN-oplossing binnen deze infrastructuur wordt een vergelijking gemaakt op basis van de Weighted Ranking Method (WRM). De volgende criteria worden daarbij in overweging genomen:

- Gebruiksvriendelijkheid: Hoe eenvoudig is het voor gebruikers om verbinding te maken met de VPN? Worden er gebruiksvriendelijke clients of configuratiewizards aangeboden?
- Prijs: Is de oplossing gratis (open-source) of zijn er kosten verbonden aan licenties of abonnementen?
- Self-hostingmogelijkheden: Kan de VPN-oplossing volledig zelf worden gehost, of is men afhankelijk van externe diensten?
- Beveiligingsopties (zoals wachtwoord + MFA): Ondersteunt de oplossing aanvullende beveiligingsopties zoals Multi-Factor Authentication (MFA), certificaatgebaseerde toegang, of rolgebaseerde toegang?

Op basis van deze criteria wordt een gewogen score berekend die helpt bij het objectief selecteren van de VPN-oplossing die het best past bij de beveiligingsbehoeften en flexibiliteitseisen van dit project.

Ease of use	OpenVPN	Wireguard	Tailscale
Gebruiksgemak	3	3	4
Prijs	2	0	3
Selfhosted	4	5	1
Password/mfa (extras)?	3	4	3

OpenVPN is relatief gebruiksvriendelijk, maar vereist enige configuratie-inzicht. Het is open source en biedt commerciële versies en ondersteuning. Een belangrijk voordeel van OpenVPN is dat het volledig zelfgehost kan worden, wat flexibiliteit biedt in de implementatie. Daarnaast ondersteunt het authenticatie met wachtwoorden en kan het worden uitgebreid met multi-factor authenticatie (MFA) via plugins, wat de beveiliging versterkt.(*Business VPN for secure networking | OpenVPN*, z.d.)

WireGuard is zeer gebruiksvriendelijk met een eenvoudige configuratie. Het is ook open source en gratis te gebruiken, en kan volledig zelfgehost worden. WireGuard ondersteunt authenticatie via publieke en private sleutels, maar heeft geen ingebouwde MFA, hoewel externe oplossingen kunnen worden geïntegreerd. (Donenfeld, z.d.)

Tailscale is de meest gebruiksvriendelijke optie met een snelle setup en configuratie via een webinterface. Het is gratis voor kleine teams, met betaalde plannen voor grotere teams en extra functies. Tailscale is echter niet volledig zelfgehost, maar kan worden gebruikt met een eigen Tailscale-server voor geavanceerde configuraties. Het biedt sterke ondersteuning voor MFA, wat het aantrekkelijk maakt voor beveiligingsbewuste gebruikers.(*Tailscale · Best VPN Service for Secure Networks*, z.d.)

Na zorgvuldige overweging hebben we gekozen voor OpenVPN vanwege de flexibiliteit van zelfhosting, de uitgebreide configuratiemogelijkheden en de ondersteuning voor multi-factor authenticatie. Tailscale hebben we als backup-optie, gezien de gebruiksvriendelijkheid en de snelle setup, wat het een waardevolle aanvulling maakt voor onze netwerkinfrastructuur.

3. Realisatie

Dit hoofdstuk bevat informatie over wat er gerealiseerd is voor dit project. Het begint bij de setup en de onderdelen die daar gebruikt worden en dan over de tools die het project bevatten.

3.1. Setup

De volledige setup is modulair opgezet. De structuur bestaat uit drie belangrijke lagen:

- Infrastructuurlaag: OpenTofu voor provisioning van virtuele machines, netwerken, firewalls en DNS.
- Configuratielaag: Cloud-init voor initiële setup en Ansible voor verdere automatisering en installatie van tooling.
- Automatisatielaag: Een Python-wrapper voor orkestratie en gebruiksgemak.

Gevoelige gegevens zoals API-tokens, IP-adressen, gebruikersnamen en wachtwoorden worden beheerd in een “.env”-bestand dat centraal wordt ingelezen door het Python-script.
De Setup wordt volledig doorlopen in bijlage1: “Readme.pdf”.

3.1.1. Python

De Python-wrapper vormt het hart van het automatiseringsproces. Deze voert de volgende taken uit:

- Inlezen van het “.env”-bestand en exporteren van variabelen naar OpenTofu en Ansible.
- Aanroepen van de OpenTofu-commando's om infrastructuur uit te rollen of te vernietigen.
- Automatisch genereren van SSH-configuratiebestanden op basis van de gegenereerde hosts.
- Health-checks uitvoeren op uitgerolde services na provisioning.
- Logging en foutafhandeling tijdens het gehele deploymentproces.

Hierdoor wordt het volledige proces met één commando uitvoerbaar, wat zowel fouten vermindert als tijd bespaart.

Hieronder is een kleine code snipped van de python code.

```python
def deploy(client, domain=None, subdomain=None, cf_api_token=None, gophish_pass=None, mythic_pass=None, ansible_pass=None, ubuntu_pass=None, location=None, zip_password=None):
    cidr, ips = generate_cidr()
    print(f"Generated CIDR for {client}: {cidr}")

    # Check if the workspace already exists and select it if it exists
    result = subprocess.run(["tofu", "workspace", "list"], cwd=tofu_DIR, capture_output=True, text=True)
    workspaces = [line.replace("*", "").strip() for line in result.stdout.strip().splitlines()]
    if client in workspaces:
        print(f"Workspace '{client}' already exists. Selecting it.")
        subprocess.run(["tofu", "workspace", "select", client], cwd=tofu_DIR, check=True)
    else:
        print(f"Creating new workspace: {client}")
        subprocess.run(["tofu", "workspace", "new", client], cwd=tofu_DIR, check=False)

    # Build the apply command with optional arguments
    load_dotenv(dotenv_path=dotenv_path)
    apply_command = [
        "tofu", "apply",
        "-var", f"client_name={client}",
        "-var", f"client_cidr={cidr}",
        "-var", f"ip_range={ips[0]}",
        "-var", f"gateway={ips[1]}",
        "-var", f"ip_proxy={ips[2]}",
        "-var", f"ip_command={ips[3]}",
        "-var", f"ip_phishy={ips[4]}",
        "-var", f"ip_attack={ips[5]}",
        "-var", f"ip_jumphost={ips[6]}",
        "-var", f"ip_evil={ips[7]}",
        "-auto-approve"
    ]
```

Figuur 1: code snipped python

3.1.2. OpenTofu

OpenTofu wordt ingezet voor declaratieve infrastructuur-uitrol. De infrastructuurcode is onderverdeeld in modulaire componenten:

- Netwerken: Subnetten en routing tussen publieke en private netwerken.
- Definiëring van alle servers (zoals proxy-server, C2-server, Exegol, GoPhish, enz.).
- Firewalls: Automatisch opgestelde firewallregels per serverrol.
- SSH-sleutels: Automatisch toegevoegd aan elke instance voor veilige toegang.
- Cloudflare-integratie: Automatisch aanmaken van DNS-records (A, CNAME, TXT) voor alle domeinen en subdomeinen die nodig zijn voor C2 en phishing.
- Provisioners: Integratie met Cloud-init voor post-boot configuratie.

Hieronder staan een paar code snippeds van de OpenTofu code.
De eerste code snipped is van de automatische dns records op cloudflare.
De volgende code snipped is voor de discord webhook die zegt wanneer de omgeving is afgebroken.

```
# A record for root domain
resource "cloudflare_dns_record" "root" {
    zone_id = data.cloudflare_zones.my_zone.result[0].id
    name    = var.domain
    type    = "A"
    content   = hcloud_server.evil.ipv4_address
    ttl     = 1
    proxied = false
}

# A record for subdomain.domain.com
resource "cloudflare_dns_record" "dashboard" {
    zone_id = data.cloudflare_zones.my_zone.result[0].id
    name    = var.subdomain
    type    = "A"
    content   = hcloud_server.proxy.ipv4_address
    ttl     = 1
    proxied = false
}

# Wildcard A record for *.domain.com
resource "cloudflare_dns_record" "wildcard" {
    zone_id = data.cloudflare_zones.my_zone.result[0].id
    name    = "*"
    type    = "A"
    content = hcloud_server.evil.ipv4_address
    ttl     = 1
    proxied = false
}
```

Figuur 2: code snipped cloudflare opentofu

```
# Discord notification resource

# This resource sends a notification to a Discord channel when the resources are destroyed.
resource "null_resource" "Discord_notification" {
    triggers = {
        client_name = var.client_name
        discord_webhook_url = var.discord_webhook_url
    }
    provisioner "local-exec" {
        when = destroy
        command = <<EOT
            curl -X POST -H 'Content-type: application/json' --data '{"content":"Hetzner Cloud resources for ${self.triggers.client_name} have been destroyed."}'
        EOT
    }
}
```

Figuur 3: code snipped discord opentofu

```
# create the proxy host with a firewall
resource "hcloud_server" "proxy" {
    name = "proxy-${var.client_name}"
    image = var.ostype
    server_type = "cx22"
    location = var.location
    ssh_keys = data.hcloud_ssh_keys.all.ssh_keys.*.name

    network {
        network_id = hcloud_network.network.id
        ip = var.ip_proxy
    }
    public_net {
        ipv4_enabled = true
        ipv6_enabled = false
    }
    user_data = templatefile("Cloud_init/cloud-init-proxy.yaml", {
    tailscale_auth_key = var.tailscale_auth_key
    cidr = var.client_cidr
    ip_phishy = var.ip_phishy
    domain=var.domain
    subdomain = var.subdomain
    ip_command = var.ip_command
    cloudflare_api_token = var.cloudflare_api_token
    zone_id = data.cloudflare_zones.my_zone.result[0].id
    playbook_proxy = (indent(6, templatefile("./ansible/playbook_proxy.yml", {
        ip_jumphost = var.ip_jumphost
        ip_command = var.ip_command
        ip_phishy = var.ip_phishy
        ip_evil = var.ip_evil
        domain = var.domain
        subdomain = var.subdomain
    })))
    })
    firewall_ids = [hcloud_firewall.vuurmuur.id]

    depends_on = [ hcloud_network_subnet.subnet ]
}
```

Figuur 4: code snipped server opentofu

```
# Network resources

# Creates a private network named "network" with an IP range.
resource "hcloud_network" "network" {
    name = "network_${var.client_name}"
    ip_range = var.client_cidr
    depends_on = [ null_resource.Discord_notification ]
}

# Defines a subnet within the created network.
# The subnet is of type "cloud" and is located in the "eu-central" network zone.
# Uses the same IP range as the network
# Depends on the successful creation of the `hcloud_network` resource.
resource "hcloud_network_subnet" "subnet" {
    network_id = hcloud_network.network.id
    type = "cloud"
    network_zone = "eu-central"
    ip_range = var.client_cidr
    depends_on = [ hcloud_network.network ]
}

# Adds a route to the network, directing traffic destined for 0.0.0.0/0 (default route) to the gateway at 10.xx.xx.2.
# Depends on the successful creation of the `hcloud_network_subnet` resource.
resource "hcloud_network_route" "weg" {
    network_id = hcloud_network.network.id
    destination = "0.0.0.0/0"
    gateway = var.ip_proxy
    depends_on = [ hcloud_network_subnet.subnet ]
}

resource "hcloud_firewall" "vuurmuur" {
    name = "vuurmuur_${var.client_name}"
    rule {
        direction = "in"
        protocol = "tcp"
```

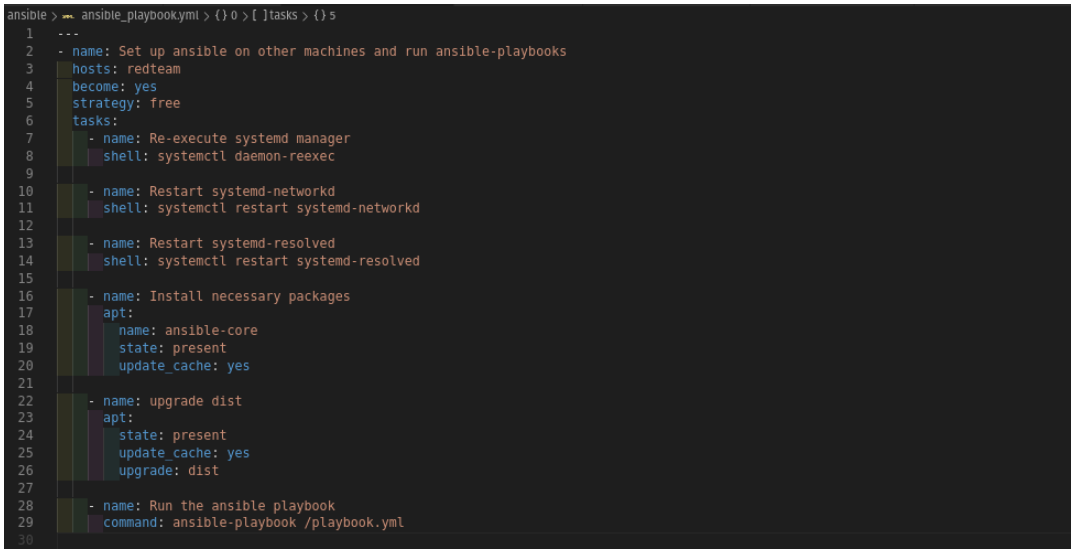
Figuur 5: code snipped network opentofu

3.1.3. Ansible

Ansible automatiseert de configuratie van alle Red Team-tools en aanvullende services. De playbooks zijn geoptimaliseerd voor herhaalbaarheid en foutafhandeling:

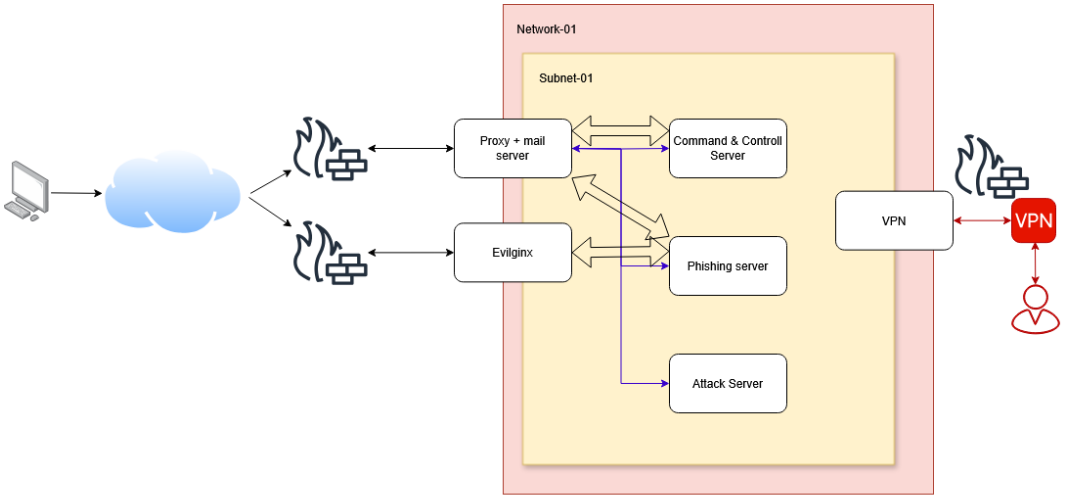
- GoPhish: Installatie, configuratie, setup van SMTP, templates en campagnes.
- Evilginx2: Installatie en configuratie voor phishingproxy's met SSL-certificaten.
- Mythic C2: Volledig geautomatiseerde installatie inclusief setup van communicatieprofielen.
- Exegol: Automatische setup van een kant-en-klare aanvalsmachine.
- OpenVPN: Serverinstallatie, gebruikersbeheer en automatische certificaatgeneratie.
- Certbot: Voor automatische SSL-configuratie via Let's Encrypt.
- Configuratie van logging-agents (alloy) om logs te verzenden naar een centrale loggingserver.
- Parsen van logs van verschillende tools voor visualisatie in Grafana.

Hieronder staat een code snipped van een ansible playbook.



Figuur 6: code snipped ansible playbook

3.2. Netwerk



Legende	
Internet access	⟷
Communications	⇔

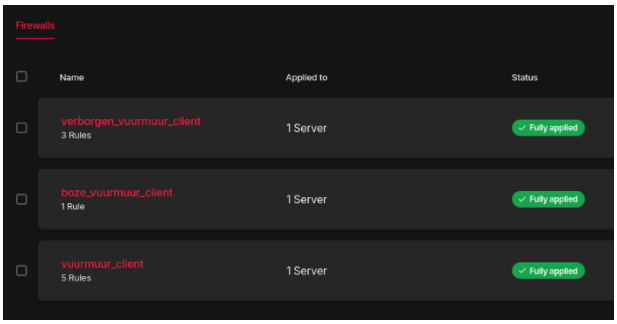
	Network	range
Network-01	10.66.X.0	/24
Subnet-01	10.66.X.0	/24

Ip-Table		
Proxy + mail	10.66.X.2	/24
Command & Controll server	10.66.X.3	/24
Phishing Server	10.66.X.4	/24
Attack Server	10.66.X.5	/24
VPN	10.66.X.6	/24
Evilginx	10.66.X.7	/24

Figuur 7: Network diagram

Hierboven zie je het netwerken diagram. Deze afbeelding kan gebruikt worden ter ondersteuning van onderstaande uitleg.
In deze netwerkinfrastructuur zijn er in totaal drie servers met een publiek IP-adres:

- De VPN-server
- De Evilginx-server
- De mail/proxy/NAT-server



Figuur 8: firewalls hetzner

Elke van deze servers is afzonderlijk beveiligd met een eigen firewall, waarbij enkel de strikt noodzakelijke poorten zijn opengesteld om de aanvalsvectoren zo beperkt mogelijk te houden.
De firewalls kan je in bijstaande figuur zien.

De mail/proxy/NAT-server fungeert als centrale gateway naar het internet. Hierdoor kunnen de interne servers, die geen publiek IP hebben, alsnog toegang krijgen tot het internet via NAT. Dit omvat de:

- Phishing server (met GoPhish)
- Command & Control (C2) server
- Attack server (Exegol)

Er is onderling verkeer tussen deze servers voor specifieke doeleinden:

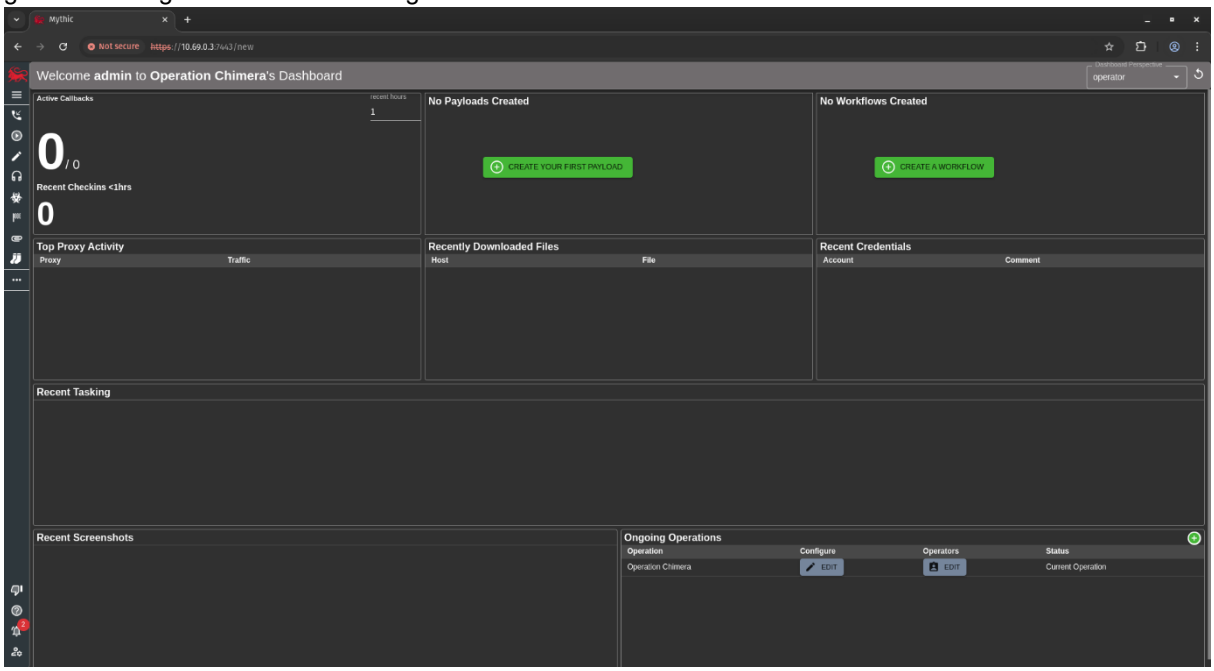
- De Phishing server maakt gebruik van Postfix op de mailserver om phishingmails te versturen.
- Voor het hosten van de phishingpagina's wordt Evilginx gebruikt. Deze intercept inloggegevens en sessiecookies.
- De Command & Control-server (Mythic) communiceert met de buitenwereld via de proxyfunctionaliteit die eveneens draait op de mail/proxy-server.

3.3. VPN

De VPN-server vormt een essentieel onderdeel van de beveiligde toegang tot de Red Team-omgeving. Binnen deze infrastructuur draait de OpenVPN-server op een eigen, geïsoleerde node om maximale veiligheid en controle te garanderen. Voor elke gebruiker worden automatisch unieke certificaten gegenereerd, waardoor toegang tot de omgeving strikt individueel en gecentraliseerd beheerd wordt. Tijdens dit proces wordt terwijl een stek password automatisch gegenereerd. Dat password dient voor de beveiliging van de configuratiebestanden. Het volledige beheer van de VPN-configuratie gebeurt via Ansible, inclusief het instellen van firewallregels, routingtabelaanpassingen en het genereren van clientconfiguraties. Toegang tot interne services zoals de Mythic C2-server en de Exegol-aanvalsmachine is uitsluitend mogelijk via de VPN, waardoor ongeautoriseerde verbindingen vanuit het publieke internet worden uitgesloten en alle communicatie end-to-end versleuteld verloopt.

3.4. Command & control

Mythic is een geavanceerd command-and-control (C2) platform en wordt centraal uitgerold binnen de Red Team-omgeving. Het platform biedt een webgebaseerde gebruikersinterface waarmee gebruikers eenvoudig payloads kunnen aanmaken en operaties kunnen beheren. Mythic ondersteunt verschillende agents zoals Apollo en Poseidon, wat het flexibel inzetbaar maakt op diverse doelplatformen zoals Windows, macOS en Linux. Daarnaast is er ondersteuning voor meerdere communicatieprofielen, waaronder HTTP(S), DNS, Slack en Discord, waarmee de communicatie tussen de C2-server en de payloads gevarieerd en stealthy kan worden ingericht. Tijdens de uitrol wordt automatisch een admin-gebruiker aangemaakt met een veilig wachtwoord.

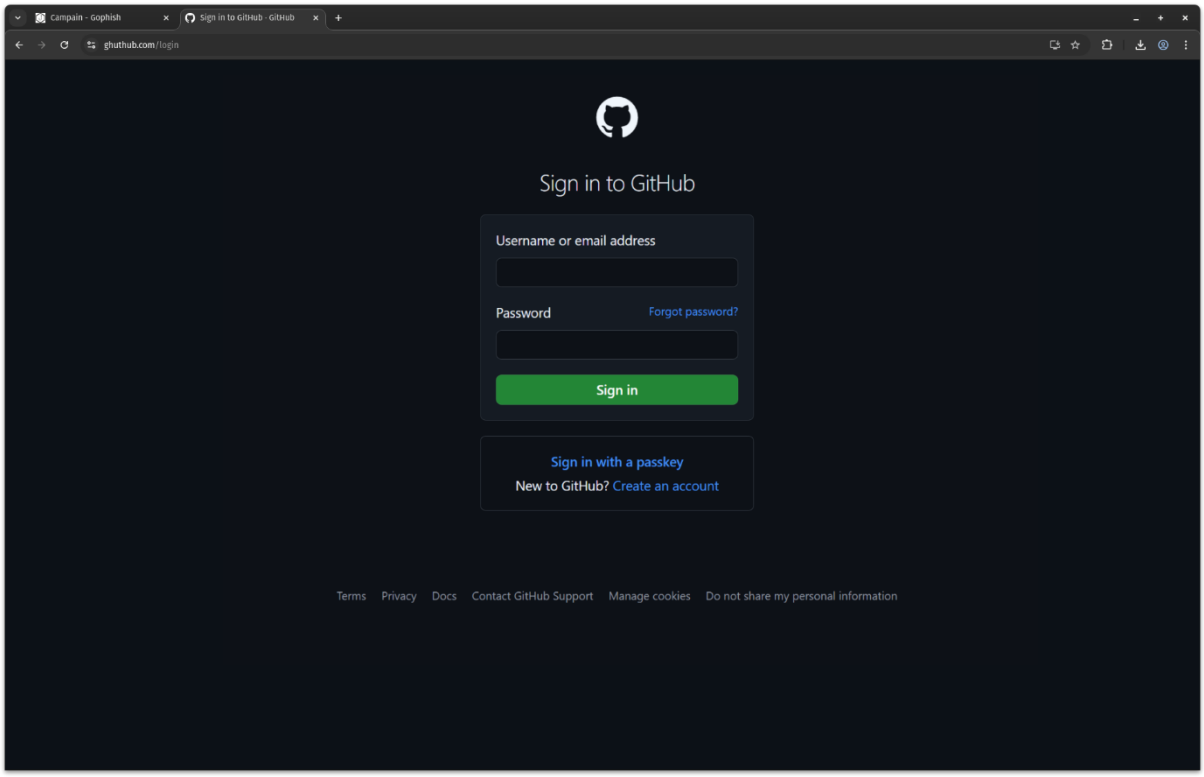


Figuur 9: screenshot mythic c2 dashboard

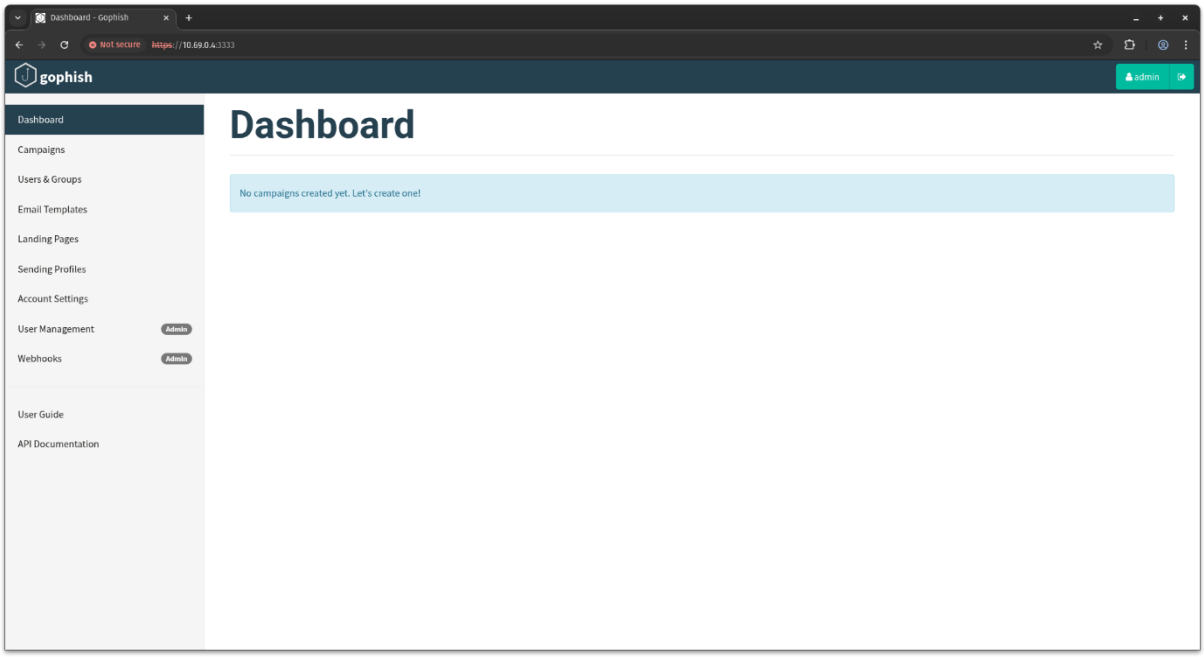
3.5. Phishing

De phishing-infrastructuur bestaat uit meerdere componenten die gezamenlijk een krachtige en realistische simulatieomgeving vormen. GoPhish wordt ingezet voor het versturen van grootschalige spear phishing-campagnes en is geïntegreerd met Postfix voor SMTP-functionaliteit, zodat gepersonaliseerde e-mails eenvoudig verzonden kunnen worden. Evilginx wordt gebruikt voor het opzetten van overtuigende phishingpagina's, inclusief mogelijkheden voor het omzeilen van tweefactorauthenticatie (2FA). Hiermee kunnen zowel inloggegevens als sessiecookies worden onderschept. In onderstaande afbeelding kan zo een fake website gezien worden. De domeinen en subdomeinen die voor deze campagnes nodig zijn,

worden automatisch aangemaakt en geconfigureerd via Cloudflare. Certbot zorgt ervoor dat alle phishingdomeinen voorzien zijn van geldige SSL-certificaten. Deze volledige setup wordt per simulatie of klantomgeving dynamisch uitgerold, wat flexibiliteit en snelheid in de operaties mogelijk maakt.



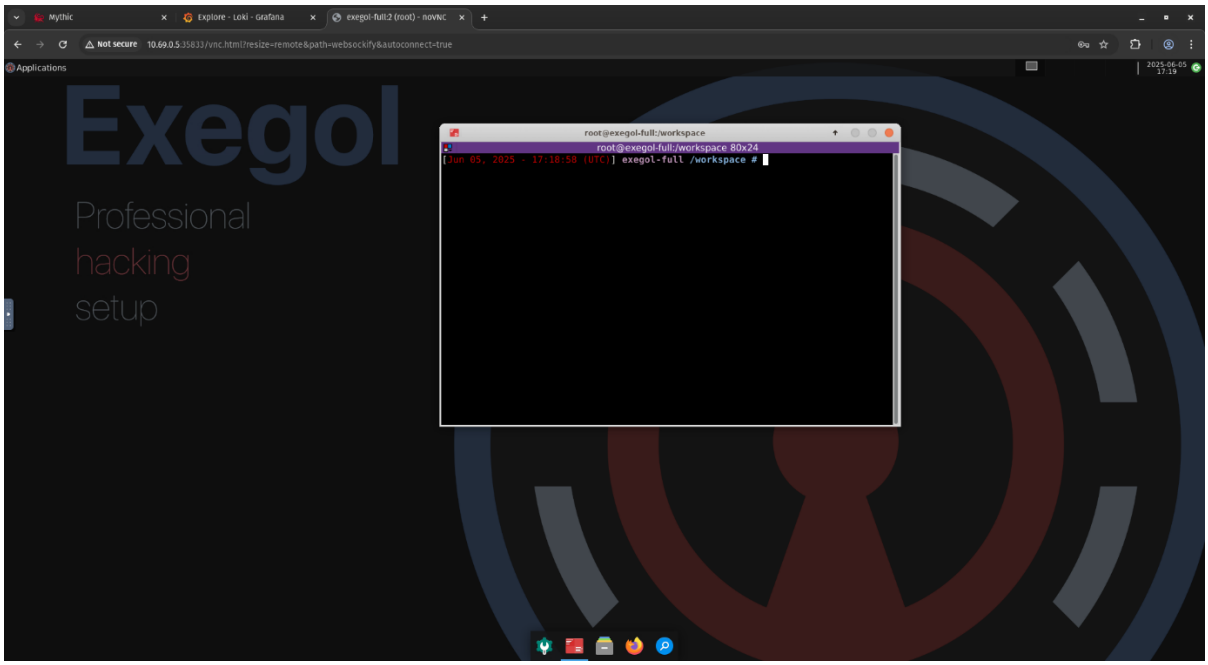
Figuur 10: Evilginx fake github page



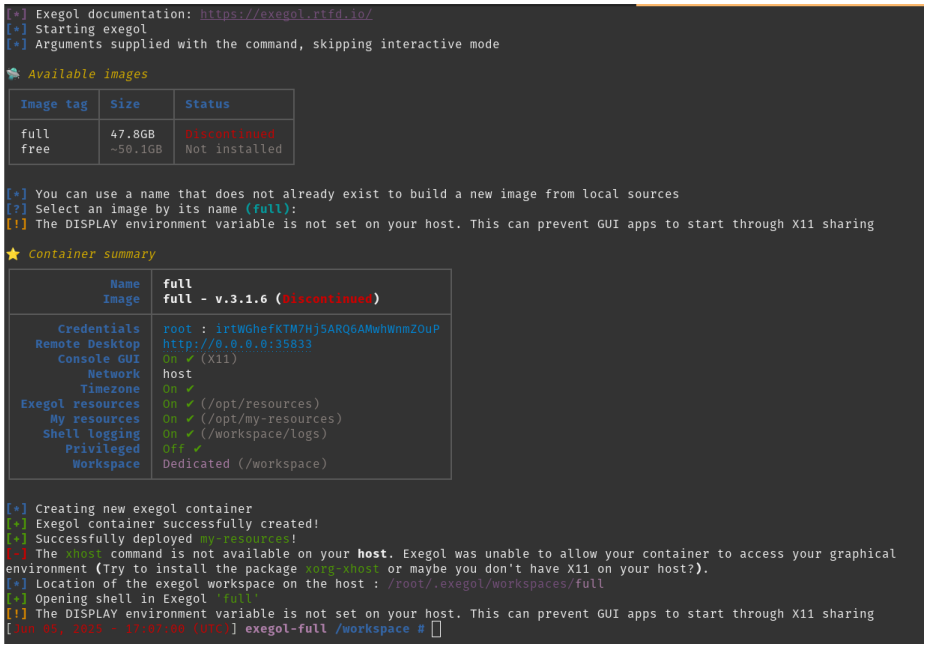
Figuur 11: Gophish dashboard

3.6. Exegol

Exegol is een krachtige, op Docker gebaseerde aanvalsmachine die speciaal ontworpen is voor gebruik in Red Team-operaties. De uitrol vindt plaats op een aparte node en gebeurt volledig geautomatiseerd via een combinatie van Cloud-init en Ansible. De machine is direct klaar voor gebruik na de installatie en bevat een uitgebreide collectie vooraf geïnstalleerde tools zoals BloodHound, CrackMapExec, Covenant en andere offensieve securitytools. Exegol is bovendien geoptimaliseerd voor gebruik met tmux en zsh, wat het werken met complexe command line-opdrachten overzichtelijk en efficiënt maakt. Logging kan standaard worden geactiveerd, zodat alle uitgevoerde handelingen kunnen worden gemonitord en bewaard voor latere analyse. In onderstaande afbeelding kan Exegol gezien worden als webapplicatie en command line.



Figuur 12: Exegol gui

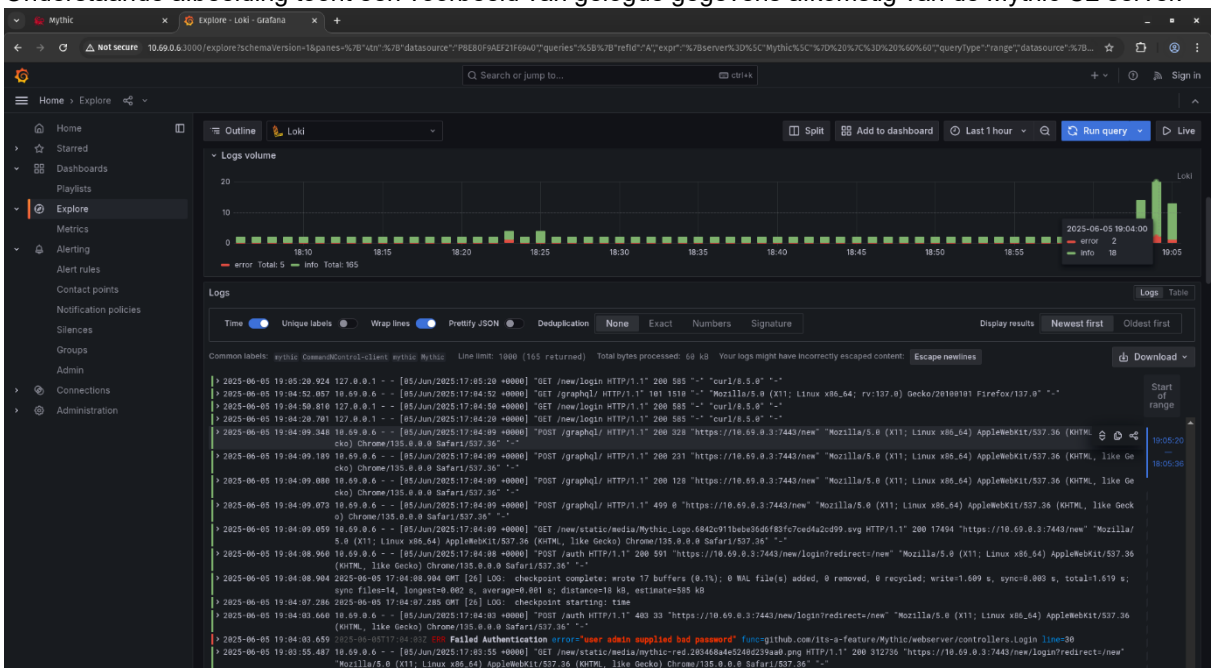


Figuur 13: Exegol cli

3.7. Logging

Voor het verkrijgen van inzicht in de werking van de omgeving en ter ondersteuning van auditing- en troubleshootingprocessen is logging een essentieel onderdeel van de infrastructuur. De visualisatie van loggegevens gebeurt via Grafana, waarbij dashboards helpen om snel afwijkingen te detecteren en de effectiviteit van operaties te monitoren.
De logs worden verzameld met behulp van Alloy en Loki, afhankelijk van de configuratie. Hierbij wordt data gegenereerd door onder andere GoPhish, Evilginx, OpenVPN, Exegol en Mythic. Deze loggegevens worden doorgestuurd naar een centrale loggingserver, waar ze worden verwerkt, opgeslagen en gearchiveerd.
Het opzetten van de logging is deels gelukt: een aantal kritieke componenten loggen correct en worden visueel weergegeven in Grafana. Door tijdsbeperkingen is het echter niet gelukt om logging voor alle onderdelen volledig te implementeren of te verfijnen. Met meer tijd zou dit verder uitgebreid kunnen worden, met meer granulariteit, betere correlatie van logs en geavanceerdere visualisaties.

Onderstaande afbeelding toont een voorbeeld van gelogde gegevens afkomstig van de Mythic C2 server.



Figuur 14: Grafana Mythic logs

4. Besluit

Met dit project werd een volledig geautomatiseerde, modulaire en schaalbare red team-omgeving gerealiseerd op Hetzner Cloud. Door gebruik te maken van moderne technologieën zoals OpenTofu, Ansible, Cloud-init en een Python-wrapper werd een infrastructuur ontwikkeld die niet alleen snel uitgerold kan worden, maar ook reproduceerbaar is over meerdere klantomgevingen. Deze omgevingen ondersteunt tools zoals Mythic C2, GoPhish, Evilginx en Exegol, en is voorzien van centrale logging en beveiligde toegang via VPN.

Uit de implementatie blijkt dat het mogelijk is om met beperkte middelen en open-source technologie een professionele red team-infrastructuur neer te zetten die zowel operationeel efficiënt als veilig is. De doelen zoals beschreven in het oorspronkelijke projectplan namelijk het automatiseren van uitrol, het reduceren van manuele handelingen en het verbeteren van herbruikbaarheid zijn succesvol behaald.

De aanpak met een Python-wrapper rondom OpenTofu bleek bijzonder krachtig, vooral door de mogelijkheid om tot 255 afzonderlijke omgevingen parallel te beheren en dit zelfs uit te breiden. Dit maakt het systeem toekomstbestendig en inzetbaar in organisaties waar meerdere red team-operaties tegelijkertijd draaien of snel moeten worden opgestart.

Voor de toekomst wordt aanbevolen om het systeem verder uit te breiden met bijvoorbeeld automatische tear-down-rapportage, integratie met CI/CD-platformen, of ondersteuning voor meer cloudproviders. Ook kan overwogen worden om compliance-controles of versleutelde dataopslag verder te automatiseren. Ten slotte is het belangrijk om waakzaam te blijven voor nieuwe beveiligingsrisico’s die kunnen ontstaan bij het gebruik van C2-omgevingen en phishingtools, zelfs in testomgevingen.

De Red Team Infrastructure zoals ontwikkeld tijdens dit project vormt een solide basis voor toekomstige red team-operaties en biedt zowel snelheid, controle als veiligheid.

Literatuurlijst

Ahmedkhlief. (z.d.). *GitHub - ahmedkhlief/Ninja: Open source C2 server created for stealth red team operations*. https://github.com/ahmedkhlief/Ninja

Ansible collaborative. (z.d.). https://www.redhat.com/en/ansible-collaborative

BlackArch Linux - Penetration Testing Distribution. (z.d.). https://blackarch.org/

Business VPN for secure networking | OpenVPN. (z.d.). https://openvpn.net/

Certbot. (z.d.). https://certbot.eff.org/

Cheap dedicated servers, cloud & hosting from Germany. (z.d.). https://www.hetzner.com/

Cloud Computing Services | Google Cloud. (z.d.). https://cloud.google.com/

Cloud computing services - Amazon Web Services (AWS). (z.d.). https://aws.amazon.com/

cloud-init 25.1.2 documentation. (z.d.). https://cloudinit.readthedocs.io/en/latest/

Cobbr. (z.d.). *GitHub - cobbr/Covenant: Covenant is a collaborative .NET C2 framework for red teamers.* https://github.com/cobbr/Covenant

Darkmidus. (z.d.). *GitHub - darkmidus/HiddenEye: Modern Phishing Tool With Advanced Functionality And Multiple Tunnelling Services [Android-Support-Available]*. https://github.com/darkmidus/HiddenEye

Docs @ Rapid7. (z.d.). https://docs.rapid7.com/

Donenfeld, J. A. (z.d.). *WireGuard: fast, modern, secure VPN tunnel*. https://www.wireguard.com/

Exegol: professional hacking setup — Exegol documentation. (z.d.). https://exegol.readthedocs.io/

Group, D. (z.d.). *Welcome! - The Apache HTTP Server project*. https://httpd.apache.org/

HAProxy - the reliable, high perf. TCP/HTTP load balancer. (z.d.). https://www.haproxy.org/

Infection Monkey | Akamai. (z.d.). https://www.akamai.com/infectionmonkey

Jenkins. (z.d.). https://www.jenkins.io/

Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution. (2025). https://kali.org/

Mythic | Mythic Documentation. (z.d.). https://docs.mythic-c2.net/

nginx. (z.d.). https://nginx.org/

Nginx Proxy Manager. (z.d.). https://nginxproxymanager.com/

OpenTofu. (z.d.). https://opentofu.org/

Overview — PoshC2. (z.d.). https://poshc2.readthedocs.io/

Parrot Security. (z.d.). https://parrotsec.org/

Saltproject.io. (z.d.). https://saltproject.io/

Server, C. W. (z.d.). *Caddy - The Ultimate Server with Automatic HTTPS*. https://caddyserver.com/

Sliver Docs: Getting Started. (z.d.). https://sliver.sh/docs?name=Getting+Started

Software, C. (2021). *Chef Software*. https://www.chef.io/

Tailscale · Best VPN Service for Secure Networks. (z.d.). https://tailscale.com/

Terraform | HashiCorp Developer. (z.d.). https://developer.hashicorp.com/terraform

Traefik, The Cloud Native Application Proxy | Traefik Labs. (z.d.). https://traefik.io/traefik/

Trustedsec. (z.d.). *GitHub - trustedsec/social-engineer-toolkit: The Social-Engineer Toolkit (SET) repository from TrustedSec - All new versions of SET will be deployed here.* https://github.com/trustedsec/social-engineer-toolkit?tab=readme-ov-file#readme

Welcome to MITRE Caldera's documentation! — caldera documentation. (z.d.). https://caldera.readthedocs.io/en/latest/

Wright, J. (z.d.). *GoPhish - Open Source Phishing Framework*. https://getgophish.com/

Bijlagen

Bijlage 1: Readme.pdf

Automated Red Team Setup

This repository provides all the resources required to deploy a robust and scalable Red Team Environment for offensive security testing and simulations. Leveraging OpenTofu for infrastructure provisioning on Hetzner Cloud, this environment is designed to emulate real-world attack scenarios in a controlled and secure setting.

Developed during an internship at Orange Cyberdefense, this project aims to deliver a comprehensive solution for red team operations, enabling advanced testing and training for cybersecurity professionals.

Overview

This environment is designed to emulate a comprehensive Red Team operation setup. It provides a secure and isolated infrastructure for staging attacks, hosting sophisticated phishing campaigns, and managing a command-and-control (C2) server. The setup enables cybersecurity professionals to conduct realistic offensive security simulations and training in a controlled environment.

Table of Contents

Team

Name	GitHub
Thomas Deboel	@Thomas
Bryan Poleunis	@Bryan

Components

- **Python Wrapper for OpenTofu**
  Simplifies the deployment process for the Red Team environment.
- **Firewalls**
  Implements strict rules to secure access to the environment.
- **Proxy and Mail Server**
  Functions as a reverse proxy and mail server for the environment.
  - **Postfix**: Handles email delivery.
  - **Nginx**: Routes traffic to the C2 server.
- **Command and Control (C2) Server**
  Hosts Mythic C2 for managing implants and payloads during engagements.
  - Accessible via a web interface.
- **Phishing Server**
  Facilitates phishing campaigns and payload hosting.
  - **GoPhish**: Manages customizable phishing campaigns.
  - Accessible via a web interface.
- **Evilginx Server**
  Enables advanced phishing campaigns and credential harvesting.
  - Accessible via CLI.
- **Attack Server**
  Equipped with Exegol, a red team attack VM/Container for executing operations against targets.
  - Accessible via CLI and noVNC (web interface).
- **Jump Server**
  Provides secure access to the environment through OpenVPN, with Tailscale as a backup.
- **.env**
  Contains environment variables for configuration, including API tokens

How to use Red Team Environment

Prerequisites

- **`.env` File**: Ensure the following variables are configured (refer to `.env.example` for guidance):
  - Discord webhook URL
  - Hetzner Cloud API token
  - Tailscale authentication key
  - repository URL for grafana docker compose file
  - GitHub token (for private repositories)

  Example `.env` Variables

```
TF_VAR_hcloud_token=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
# required

TF_VAR_cloudflare_api_token=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
# optional

TF_VAR_tailscale_auth_key=tskey-auth-xxxxxxxxxxxxxxxxxxx-xxxxxxxxxx
# required

TF_VAR_discord_webhook_url=https://discordapp.com/api/webhooks/xxxxx
# required

TF_VAR_github_token=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
# required

TF_VAR_grafana_docker=https://raw.githubusercontent.com/user/repo/re
# required (for grafana)
```

Deployment

A Python wrapper for OpenTofu to simplify the deployment process. You can deploy max 255 Environments.

Before you do anything

```
python3 deploy.py init
```

Usage

```
python3 deploy.py [-h] {init,deploy,list,destroy,destroy-all} ...
```

Manage workspaces for the Red Team environment.

Positional Arguments

- `{init,deploy,list,destroy,destroy-all}`
  Available commands:
  - `init`
    initialize the Tofu and the environment.
  - `deploy <client>`
    Deploy a new workspace for the specified client.
  - `list`
    List all workspaces and their CIDRs.
  - `destroy <client>`
    Destroy the specified client's workspace.
  - `destroy-all`
    Destroy all workspaces (requires confirmation).

Options

- `-h, --help`
  Show this help message and exit.

Subcommand Help

Init Command

```
python3 deploy.py init
```

Initialize the Tofu and the environment.
(behind the scenes it will run `tofu init`)

Deploy Command

```
python3 deploy.py deploy [-h] [--domain DOMAIN] [--subdomain SUBDOMAIN] |
```

- **Positional Arguments:**
  - `client`
    Name of the client.
- **Options:**

- `-h, --help`
  Show this help message and exit.
- `--domain DOMAIN, -d DOMAIN`
  Full domain name.
- `--subdomain SUBDOMAIN`
  Subdomain for proxy and mail.
- `--cloudflare_api_token CLOUDFLARE_API_TOKEN`
  Cloudflare API token (40 chars).
- `--gophish_password GOPHISH_PASSWORD`
  Password for GoPhish.
- `--mythic_password MYTHIC_PASSWORD`
  Password for Mythic.
- `--ansible_password ANSIBLE_PASSWORD`
  Password for Ansible user.
- `--ubuntu_password UBUNTU_PASSWORD`
  Password for Ubuntu user.
- `--zip_password ZIP_PASSWORD, -zp ZIP_PASSWORD`
  Password for the zip file with passwords for the VPN's.
- `--location {nbg1,hel1,fsn1,ash,hil,sin}, -l {nbg1,hel1,fsn1,ash,hil,sin}`
  Location for deployment:
  - `nbg1` (Nuremberg)
  - `hel1` (Helsinki)
  - `fsn1` (Falkenstein)
  - `ash` (Ashburn)
  - `hil` (Hillsboro)
  - `sin` (Singapore)

example:
make sure that the domain is in the Cloudflare account where you have the API token for.

```
python3 deploy.py deploy --domain domain.com --subdomain sub --gophish_pa
```

List Command

```
python3 deploy.py list [-h]
```

- **Options:**

- `-h, --help`
  Show this help message and exit.

Destroy Command

```
python3 deploy.py destroy [-h] <client>
```

- **Positional Arguments:**
  - `client`
    Name of the client to destroy.
- **Options:**
  - `-h, --help`
    Show this help message and exit.

Destroy-All Command

```
python3 deploy.py destroy-all [-h]
```

- **Options:**
  - `-h, --help`
    Show this help message and exit.

VPN

To access the environment, you need to connect to the VPN. The VPN is configured using OpenVPN. You can use Tailscale as a backup.
To connect to the VPN, follow these steps:

Linux

1. Install OpenVPN on your local machine.
2. Download the OpenVPN configuration file from the environment.
3. Download the password file (it will be in base64-encoded ZIP format).
4. Decode the password file using the following command:
   `base64 -d <password.zip> > <password_decoded.zip>`
5. Unzip the password file using the following command:

```
unzip -P <password> <password_decoded.zip>
```

6. Open a terminal and navigate to the directory where the configuration file is located.
7. Run the following command to connect to the VPN:
   ```
   sudo openvpn <file.ovpn>
   ```
8. Enter your username and password when prompted (username is the name of the file and password is in the password.zip file).

Windows

1. Install OpenVPN on your Windows machine.
2. Download the OpenVPN configuration file from the environment.
3. Download the password file (it will be in base64-encoded ZIP format).
4. Decode the password file using a tool like CyberChef or a PowerShell command:

   ```
   [IO.File]::WriteAllBytes("password_decoded.zip", [Convert]::FromBase
   ```

5. Extract the decoded ZIP file using Windows Explorer or a tool like 7-Zip.
6. Open the extracted file to find your username and password.
7. Open the OpenVPN GUI, right-click the tray icon, and select "Import file" to import the configuration.
8. Right-click the tray icon again and select "Connect".
9. When prompted, enter the username and password from the extracted file to connect to the VPN.
10. Install OpenVPN on your local machine.
11. Download the OpenVPN configuration file from the environment.
12. Open the OpenVPN GUI and right-click on the tray icon.
13. Select "Import file" and choose the configuration file.
14. Right-click on the tray icon again and select "Connect" to connect to the VPN.

macOS

1. Install Tunnelblick (recommended OpenVPN client for macOS).
2. Download the OpenVPN configuration file from the environment.

3. Download the password file (it will be in base64-encoded ZIP format).
4. Decode the password file using the Terminal:

```
base64 -d password.zip > password_decoded.zip
```

5. Extract the decoded ZIP file using Finder or the `unzip` command:

```
unzip -P password password_decoded.zip
```

6. Open the extracted file to find your username and password.
7. Double-click the `.ovpn` configuration file to import it into Tunnelblick.
8. Launch Tunnelblick and connect using the imported configuration.
9. When prompted, enter the username and password from the extracted file to connect to the VPN.

Exegol

SSH into the attack server using OpenVPN. Once connected, you can use Exegol to run various offensive security tools and scripts.

- `exegol start full --desktop --desktop-config http:0.0.0.0 -l --log-method script`
  starts a webserver on a random port with a desktop environment. You can access the desktop via a web browser and the VPN.
  - username: `root`
  - password: `password automaticly generated`

Explanation of the full command:

- `exegol start full`
  - starts Exegol with all tools and a desktop environment.
- `--desktop`
  - starts a desktop environment.
- `--desktop-config http:0.0.0.0`
  - starts a webserver on a random port with a desktop environment accessible

for anyone (in the internal network).

- `-l --log-method script`
    - logs all commands to a file in the (`./.exegol/worspaces/full/logs/`).

For detailed setup, configuration, and usage instructions, refer to the official exegol documentation.

GoPhish

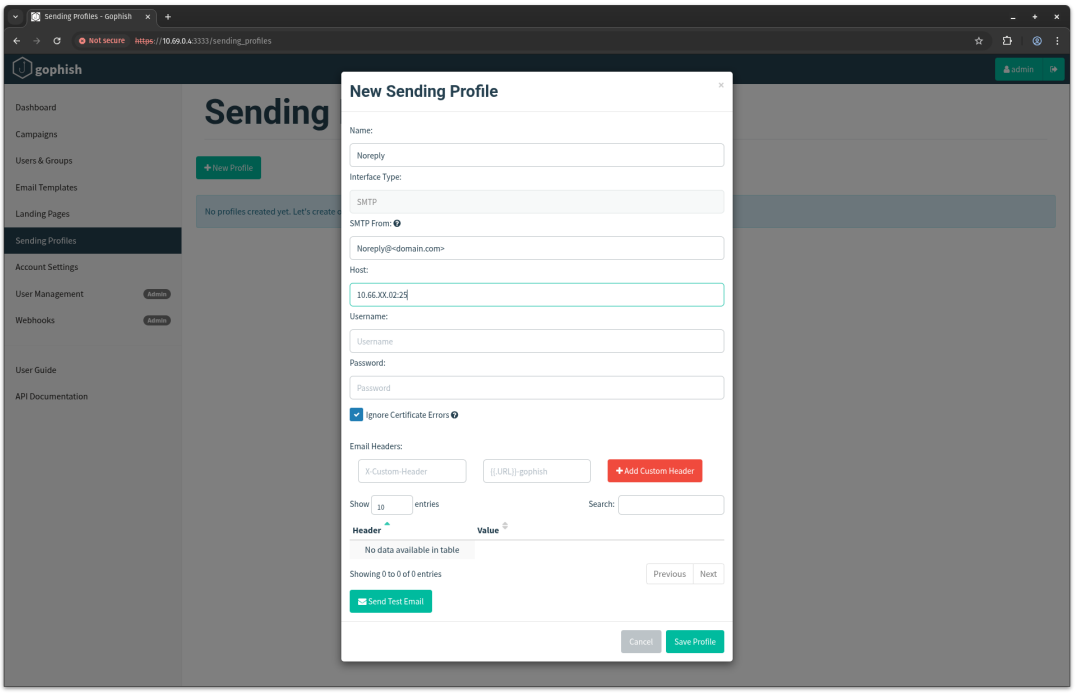
To access GoPhish, navigate to the web interface using the provided URL. The default credentials are:

- Username: `admin`
- Password: `password(in the .env file)`

Once logged in, you have to change the password. Afterwards you can create and manage phishing campaigns, including email templates, landing pages(for landing page we will use Evilginx), and tracking statistics.

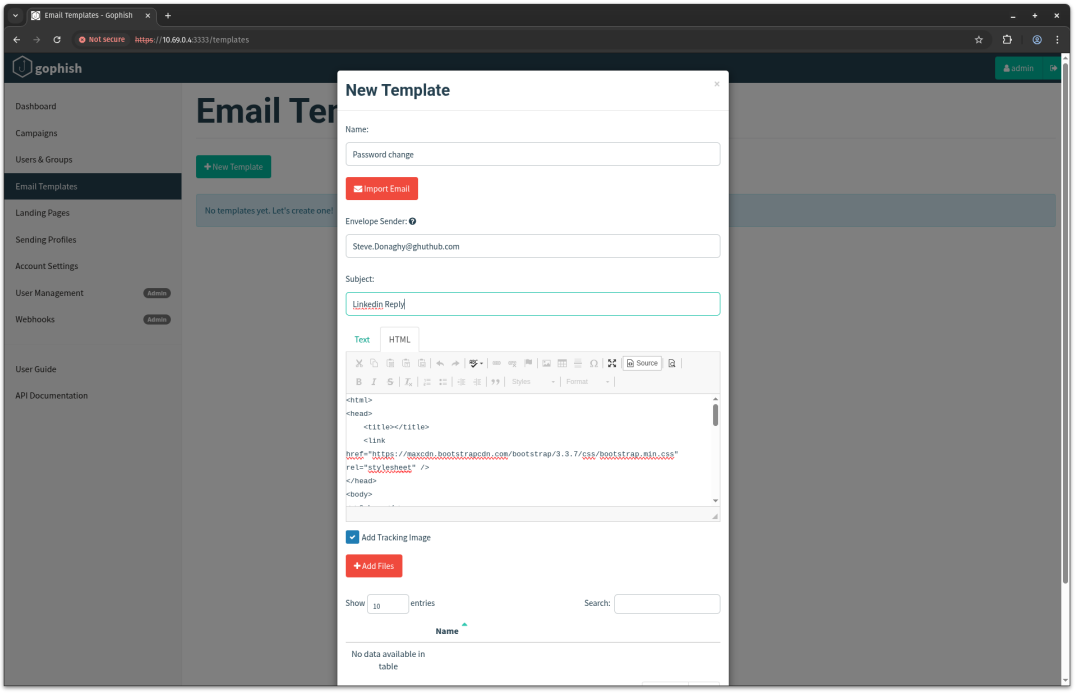
To create a new phishing campaign, follow these steps:

1. Configure sending profiles.
    - Click on "Sending Profiles" in the left sidebar.
    - Click on the "New Profile" button.
    - Fill in the profile details, including the name, email address, and SMTP server settings.
    - Click "Save" to create the sending profile.



2. Configure Email Templates.

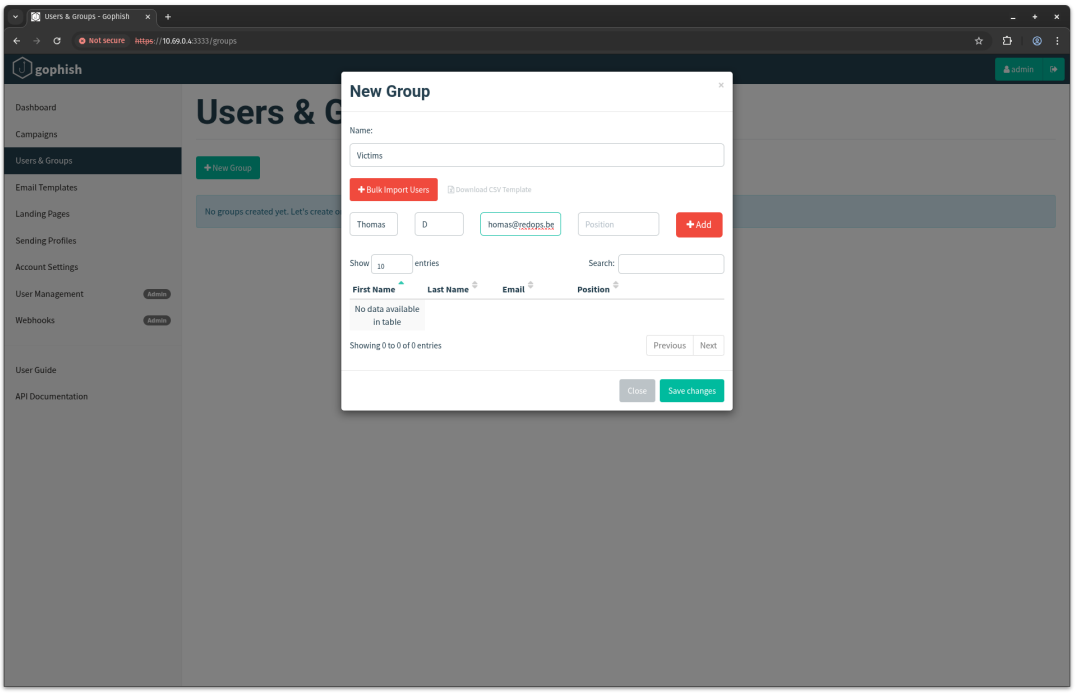
   - Click on "Email Templates" in the left sidebar.
   - Click on the "New Template" button.
   - Fill in the template details, including the name, subject, and body of the email.
     [Mail Templates]
   - Click "Save" to create the email template.



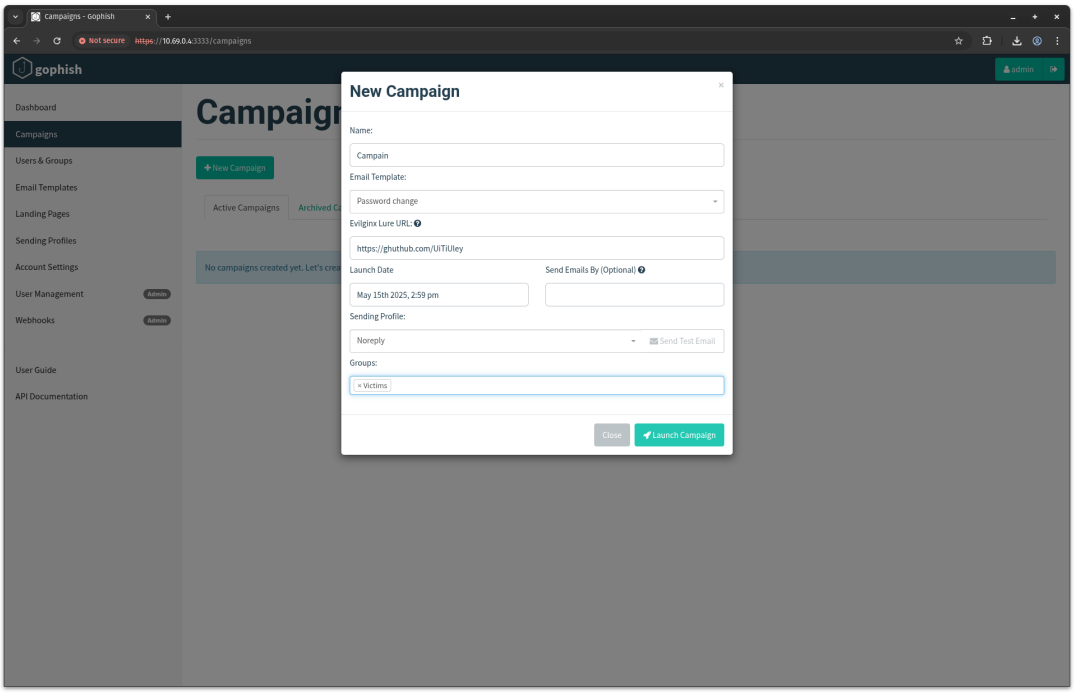
3. Configure user groups.

   - Click on "User Groups" in the left sidebar.

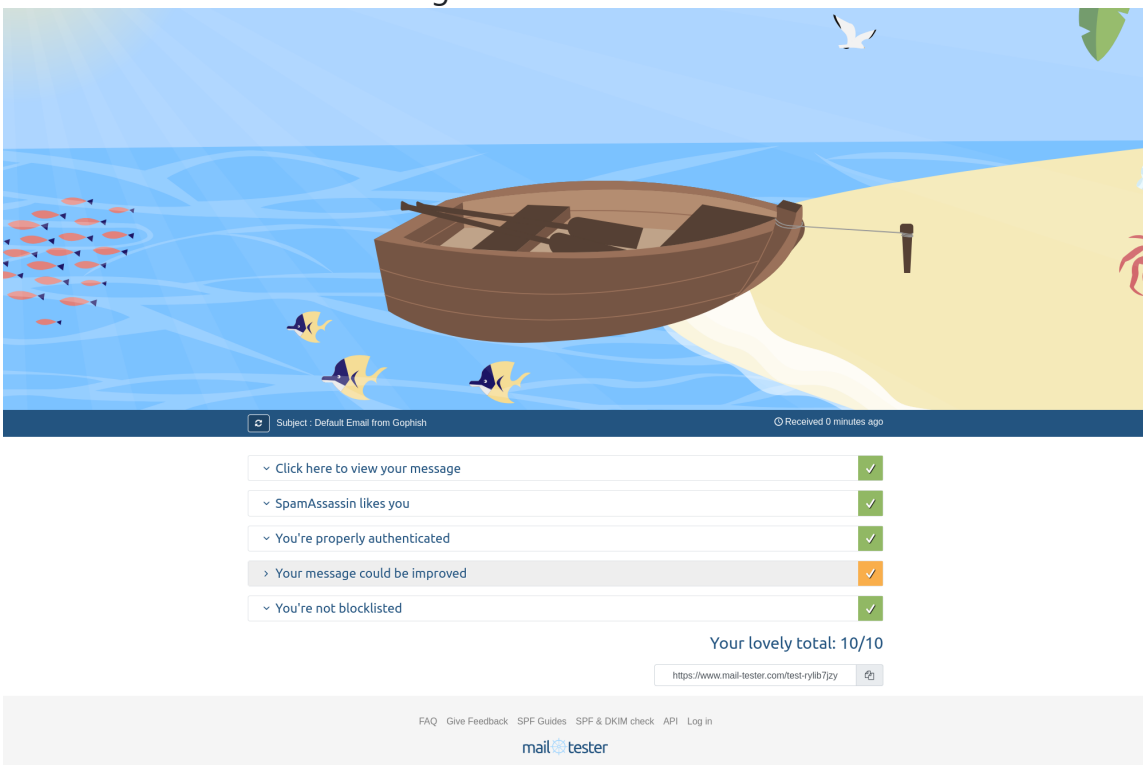
- Click on the "New Group" button.
- Fill in the group details, including the name and email addresses of the users.
- Click "Add" to add the user.
- alternatively you can use the import function to import a CSV file with the users.
- Click "Save changes" to save the user group.



4. campaigns
   - Click on "Campaigns" in the left sidebar.
   - Click on the "New Campaign" button.
   - Fill in the campaign details, including the name, email template, and landing page (for the landing page, refer to the Evilginx section and write down the lure in landing page).
   - Configure the target group and schedule the campaign.
   - Click "Launch Campaign" to start the phishing campaign.



Proof of valid mail server configuration:



Evilginx

SSH into the phishing server using OpenVPN. Once connected, you can use Evilginx to set up advanced phishing campaigns. To start Evilginx follow these steps:

Make sure you are in the Evilginx directory.

- `cd ./evilginx`

tmux

Start evilginx (with logging (tmux))

- `tmux new -s evilginx`
- `./build/evilginx -p ./phishlets -t ./redirectors | tee /var/log/evilginx.log`
  You can detach from the session by pressing `Ctrl + b` followed by `d`. To reattach to the session, run:
- `tmux attach -t evilginx`
- `tmux ls` (shows all tmux sessions)

screen

start evilginx(with logging (screen))

- `screen -S evilginx`
- `./build/evilginx -p ./phishlets -t ./redirectors | tee /etc/alloy/evilginx.log`
  You can detach from the session by pressing `Ctrl + a` followed by `d`. To reattach to the session, run:
- `screen -r evilginx`
- `screen -ls` (shows all screen sessions)

no logging

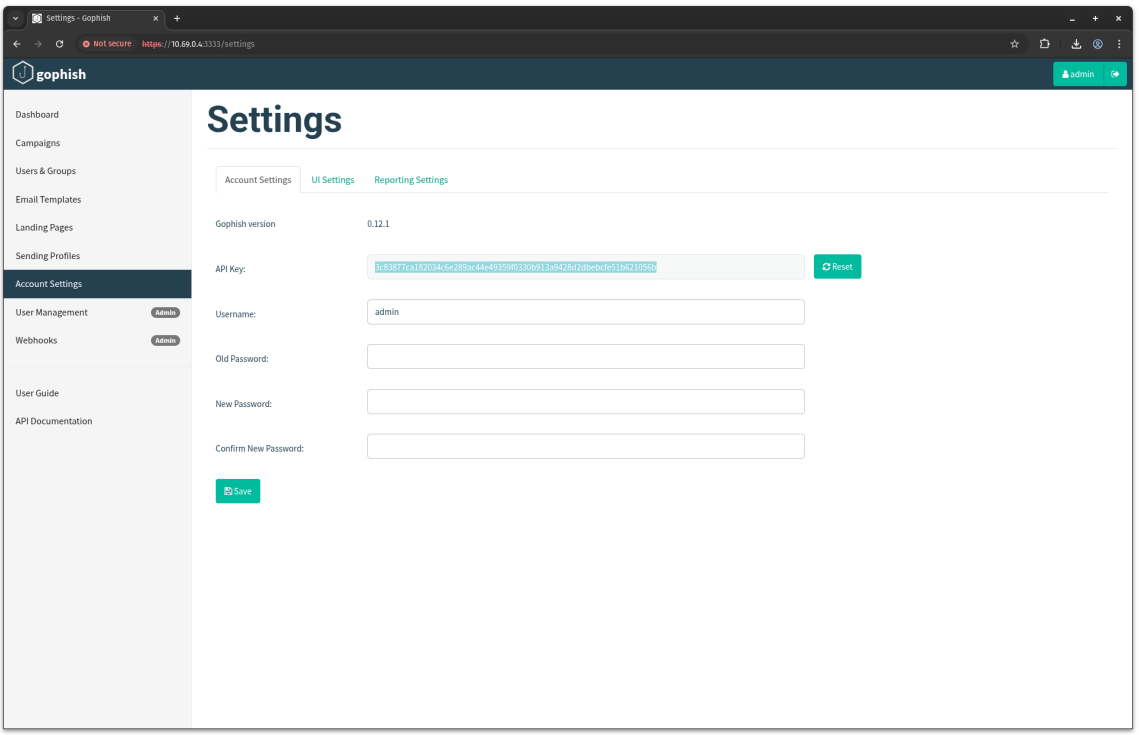
- `./build/evilginx -p ./phishlets -t ./redirectors`
  This will start Evilginx without logging.

config

To configure Evilginx:

- `config domain <domain>`
  - example: `config domain example.com`
- `config ipv4 external <external_ipv4_address>`
  - example: `config ipv4 external 192.168.10.9`
- `config gophish admin_url <gophish_admin_url>`
  - example: `config gophish admin_url https://10.69.0.4:3333`
-

`config gophish api_key <gophish_api_key>`



- `config gophish insecure true`
- `config gophish test`

Now we can start the phishlet.

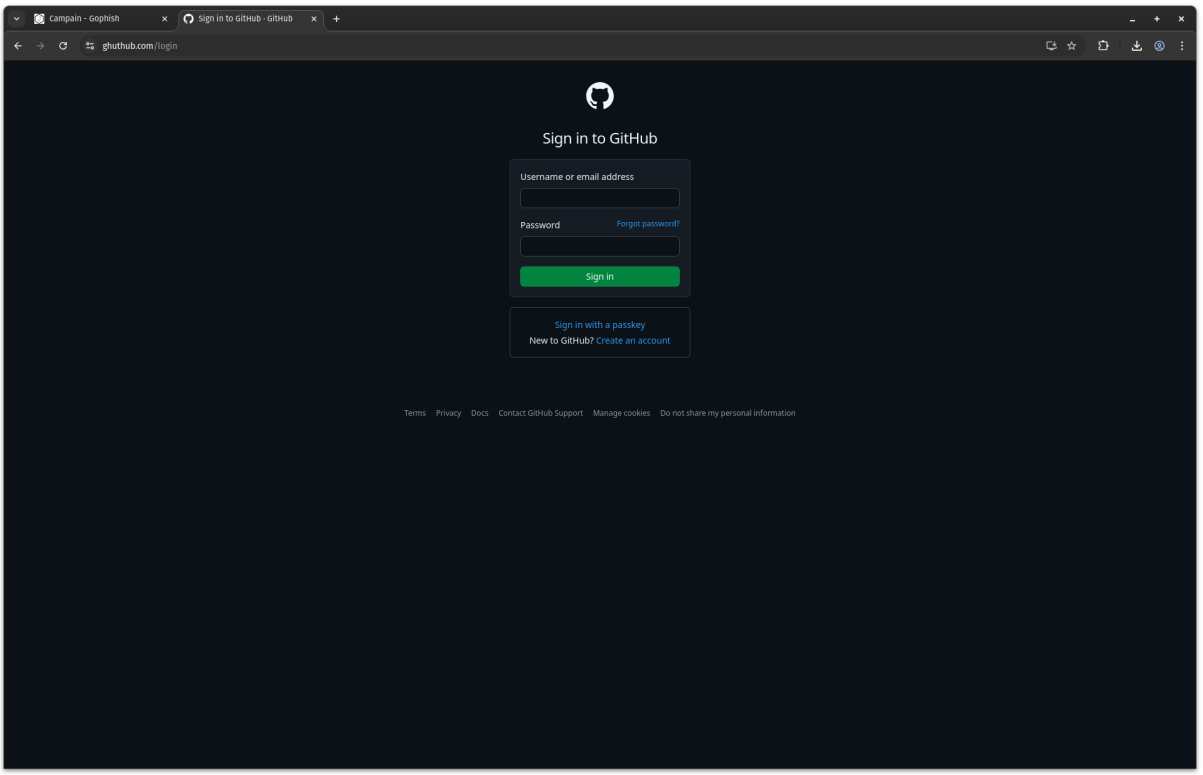
1. `phishlets hostname <phishlet_name> <domain>`
   - example: `phishlets hostname github example.com`
2. `phishlets enable <phishlet_name>`
   - example: `phishlets enable github`
3. `lures create <phishlet_name>`
   - example: `lures create github`
4. `lures` (shows all created lures)
5. `lures get-url <lure_id>` (shows the lure url)
   (This url we can use in GoPhish as the landing page url)

To view the captured sessions.

- `sessions` (shows all captured sessions)
- `sessions <session_id>` (shows the session details)
  To use the captured session download storageAce addon in Chrome and copy the session id and paste it in the storageAce.

And now you have a phishing website connected to Gophish and Evilginx

For detailed setup, configuration, and usage instructions, refer to the official Evilginx documentation.



Mythic

The domain for Mythic is automatically configured using OpenTofu.

To access Mythic, open the provided web interface URL. The default credentials are:

- **Username:** `admin`
- **Password:** The value specified in your `.env` file or provided via the command line.

Once logged in, you can create and manage implants, payloads, and tasks. Mythic offers a comprehensive platform for orchestrating red team operations.

Initial Setup

Before generating payloads, ensure your callback domain is correctly configured. Update the relevant DNS record in Cloudflare to point to your server's IP address.

Creating a Payload

1. Navigate to **Payloads** in the left sidebar.
2. Click **New Payload**.
3. Enter the required details, such as name, payload type, and configuration

options.
4. Assign the payload to the appropriate target group.
5. Click **Launch Payload** to generate and deploy it.

After launching, you can monitor payload status, view logs, and manage tasks directly from the Mythic web interface. The platform allows you to send commands, receive responses, and oversee all aspects of your red team engagement.

For more detailed information and advanced usage, refer to the Mythic Documentation.

Requirements

To set up and use the Red Team Environment, ensure you have the following:

Software and Tools

- **OpenTofu**: Version 1.9 or higher.
  - Provider requirements:
    - `hashicorp/null` v3.2.3 or higher
    - `hashicorp/hetzner` v1.50.0 or higher
    - `hashicorp/cloudflare` v5.3.0 or higher
    - `hashicorp/random` v3.7.2 or higher
- **Python**: Version 3.10 or higher, with `pip` version 24.2 or higher.
- **Python Libraries**: Install the required libraries using the following command: `pip install -r requirements.txt`
- **OpenVPN**: Required for VPN access (Tailscale can be used as a backup).

Additional Notes

- **Install Dependencies**: Verify that all required software and libraries are installed and properly configured before starting deployment.
- **Secure the `.env` File**: The `.env` file contains sensitive credentials and configuration values—store it securely and restrict access.
- **Domain Requirements**: A valid domain is necessary for configuring phishing campaigns and the C2 server.
- **Familiarity Recommended**: Basic knowledge of the tools and technologies described above will help you use the environment effectively.

Future Improvements

Planned Improvements

The following enhancements are planned to further improve the Red Team Environment:

Logging & Monitoring

- **Enhanced Log Readability**
  - Improve formatting and structure of Exegol logs for easier analysis.
  - Refine Evilginx log output for better clarity and troubleshooting.
- **Automated Log Handling**
  - Automatically archive logs (e.g., in ZIP format) after teardown.
  - Send archived logs to collaboration platforms such as Discord, Microsoft Teams, or Slack for centralized review.

VPN & Security

- **VPN Hardening**
  - Strengthen VPN configuration for improved security (e.g., stronger ciphers, stricter access controls).
  - Implement Two-Factor Authentication (2FA) for VPN access to enhance user authentication.
- **Environment Encryption**
  - Encrypt sensitive environment data at rest and in transit to protect against unauthorized access.

Multi-Cloud Support

- **Additional Cloud Providers**
  - Extend support to other cloud platforms such as AWS, Azure, and Google Cloud, enabling flexible deployment options.

Threat Simulation & Response

- **Decoy Services (Honeypots)**
  - Integrate honeypot or decoy services to detect and analyze unauthorized activity within the environment.
- **Real-Time Credential Alerts**
  - Instantly forward captured credentials to security channels (Discord, Teams, Slack) for immediate awareness and response.

Environment Lifecycle

- **Improved Teardown Process**

- Streamline environment destruction, ensuring all resources are securely removed and relevant data is archived and reported.

Have suggestions or want to contribute? Open an issue or pull request!

development

to run the development version of the wrapper, you can use the following command:

```
dotenv -e /bin/bash
tofu init
tofu plan
tofu apply
```